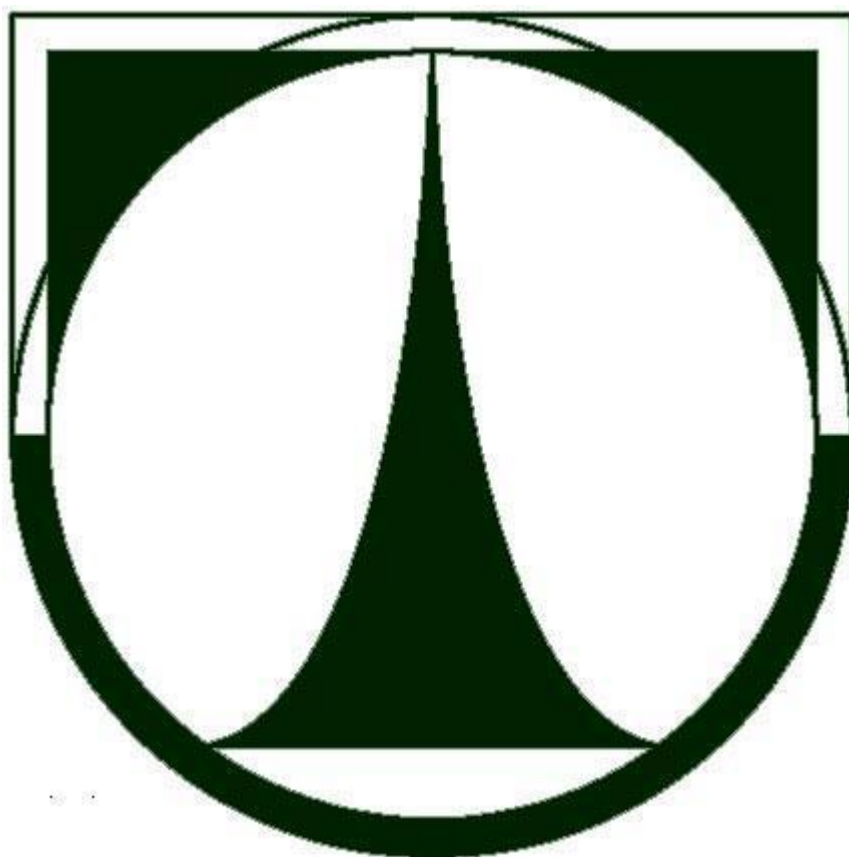


**TECHNICKÁ UNIVERZITA V LIBERCI**

**Ekonomická fakulta**



**Bakalářská práce**

**2013**

**Jan Rudolf**

# **TECHNICKÁ UNIVERZITA V LIBERCI**

## **Ekonomická fakulta**

Studijní program: B 6209 Systémové inženýrství a informatika

Studijní obor: Manažerská informatika

### **Rozšíření informačního systému Unicorn Universe s použitím metodiky Unicorn Enterprise System Powered Company**

**Expansion of Unicorn Universe information system by Unicorn  
Enterprise System Powered Company methodology**

**BP-EF-KIN-2013-21**

Autor: **Jan Rudolf**

Vedoucí práce: Ing. Vladimíra Zádová, Ph.D, Katedra informatiky

Konzultant: Ing. Zuzana Líznerová, Unicorn a. s.

Počet stran: 57

Počet příloh: 2





## **Prohlášení**

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

V Liberci, 10. 05. 2013

## **Poděkování**

Rád bych zde poděkoval vedoucí mé bakalářské práce Ing. Vladimíře Zádové, Ph.D. za odborné rady, věcné připomínky, ochotu a vstřícný přístup během konzultací i zpracování této práce.

Mé poděkování patří nemenší měrou i Ing. Zuzaně Líznerové ze společnosti Unicorn a. s. za ochotné jednání, spolupráci, odborné konzultace a poskytnutí informací o společnosti.

Velké poděkování náleží celé mé rodině za podporu, trpělivost a povzbuzování po dobu mého studia a za obecný pohled a konstruktivní kritiku této bakalářské práce.

## **Anotace a klíčová slova**

Tato práce je zaměřena na informační systém Unicorn Universe a jeho užití pro rozšíření subsystému Unicorn College. Hlavním cílem bakalářské práce je návrh, dokumentace a implementace aplikací do subsystému Unicorn College na základě požadavků na usnadnění publikace studijních výsledků a automatizace zápisu předmětů do akademického roku.

Nedílnou součástí této bakalářské práce je seznámení s metodikou Unicorn Enterprise System Powered Company, architekturou a nástroji používanými v rámci Unicorn Universe.

Unicorn College je soukromá vysoká škola využívající ke svému běhu informační systém Unicorn Universe. Rozšíření tohoto systému bylo řešeno v rámci dvou semestrů během řízené praxe ve firmě Unicorn a.s.

### **Klíčová slova**

Informační systém, Unicorn Universe, Unicorn College, školní informační podpora

## **Annotation and key words**

This work is aimed at information system Unicorn Universe and its use in expanding of Unicorn College subsystem. The main objective of the Bachelor thesis is concept, documentation and implementation of applications in the Unicorn College subsystem based on requests of simplifying publication of studying results and automatization of subject registration to academic year.

Integral part of the work is introduction to methodology Unicorn Enterprise System Powered Company, architecture and instruments used in the frame of Unicorn Universe.

Unicorn College is private college which uses Unicorn Universe information system as its main information support. Expansion of this information system was solved during two-semester-long directed internship at the Unicorn a.s. company.

## **Key words**

Information system, Unicorn Universe, Unicorn College, school information support



# Obsah

<b>1</b>	<b>INFORMAČNÍ SYSTÉM UNICORN UNIVERSE .....</b>	<b>13</b>
1.1	SEKTORY A TERITORIA.....	14
1.1.1	<i>My Territory</i> .....	14
1.1.2	<i>Business Territory</i> .....	15
1.2	ORGANIZAČNÍ JEDNOTKA .....	15
1.3	ROLE .....	15
1.3.1	<i>Přístupové role</i> .....	16
1.3.2	<i>Pracovní role</i> .....	17
1.4	ARTEFAKT .....	20
1.4.1	<i>Agregované objekty</i> .....	21
1.4.2	<i>Životní cyklus</i> .....	23
1.5	UNICORN ES UNIFORM RESOURCE IDENTIFIER.....	26
1.6	METODICKÝ ARTEFAKT .....	27
1.7	AUTORIZAČNÍ PRAVIDLA .....	28
1.7.1	<i>Stupeň prověření a utajení</i> .....	29
1.7.2	<i>Implicitní a explicitní práva</i> .....	29
<b>2</b>	<b>NÁSTROJE UNICORN .....</b>	<b>30</b>
2.1	UNICORN ENTERPRISE SYSTEM POWERED COMPANY .....	30
2.2	UNICORN UNIFIED BUSINESS MODELLING LANGUAGE .....	32
<b>3</b>	<b>ROZŠÍŘENÍ A AUTOMATIZACE TERITORIA UNICORN COLLEGE .....</b>	<b>35</b>
3.1	PUBLIKACE STUDIJNÍCH VÝSLEDKŮ.....	35
3.1.1	<i>Analýza současného stavu</i> .....	35
3.1.2	<i>Implementace projektu</i> .....	36
3.2	AUTOMATIZACE ZÁPISU PŘEDMĚTŮ DO AKADEMICKÉHO ROKU.....	39
3.2.1	<i>Analýza současného stavu</i> .....	39
3.2.2	<i>Implementace projektu</i> .....	40
3.2.3	<i>Zápis předmětů</i> .....	42
3.2.4	<i>Korekce zápisu</i> .....	46
<b>4</b>	<b>ZHODNOCENÍ PŘÍNOSU .....</b>	<b>49</b>
4.1	PUBLIKACE STUDIJNÍCH VÝSLEDKŮ.....	49
4.2	AUTOMATIZACE ZÁPISU PŘEDMĚTŮ.....	50

## Seznam obrázků

OBRÁZEK 1-1: STRUKTURA PLATFORMY UES.....	13
OBRÁZEK 1-2: PŘÍKLAD TERITORIÍ A ROLÍ.....	17
OBRÁZEK 1-3: PŘÍKLAD KOMPETENCE ROLÍ.....	18
OBRÁZEK 1-4: HLAVNÍ PARAMETRY ARTEFAKTU.....	20
OBRÁZEK 1-5: TYPY AGREGOVANÝCH OBJEKTŮ.....	22
OBRÁZEK 1-6: STAVY ARTEFAKTU SOFTWAREVÝ PROJEKT.....	23
OBRÁZEK 1-7: ŽIVOTNÍ CYKLUS POŽADAVKU.....	25
OBRÁZEK 2-1: 12 PROCESŮ METODIKY UESPC.....	32
OBRÁZEK 2-2: IKONY UUBML.....	33
OBRÁZEK 2-3: RELAČNÍ KONEKTORY UUBML.....	34
OBRÁZEK 3-1: SCHÉMA PROCESU PUBLIKACE VÝSLEDKŮ.....	37
OBRÁZEK 3-2: STRUKTURA VSTUPNÍHO SOUBORU S HODNOCENÍM.....	38
OBRÁZEK 3-3: LIST STUDIJNÍ VÝSLEDKY.....	39
OBRÁZEK 3-4: ŘÍDÍCÍ PANEL ZÁPISU PŘEDMĚTŮ.....	42
OBRÁZEK 3-5: SCHÉMA GENEROVÁNÍ ARTEFAKTŮ K ZÁPISU.....	43
OBRÁZEK 3-6: SCHÉMA ZÁPISU DO ZIMNÍHO SEMESTRU.....	45
OBRÁZEK 3-7: SCHÉMA KOREKCÍ ZÁPISU.....	47
OBRÁZEK 3-8: SCHÉMA ZRUŠENÍ ZÁPISU PŘEDMĚTU.....	48

## Seznam vzorců

VZOREC 1: OBECNÁ STRUKTURA URI.....	26
VZOREC 2: PROSTÁ FORMA UESURI.....	26
VZOREC 3: NORMALIZOVANÁ FORMA UESURI.....	27

## Seznam tabulek

TABULKA 1 – NÁKLADNOST PROJEKTU PUBLIKACE STUDIJNÍCH VÝSLEDKŮ .....	49
TABULKA 2 – NÁKLADY NA MANUÁLNÍ PUBLIKACI VÝSLEDKŮ .....	49
TABULKA 3 – NÁKLADNOST PROJEKTU AUTOMATIZACE ZÁPISU PŘEDMĚTŮ.....	51
TABULKA 4 – NÁKLADY NA SERVISNÍ ZÁPIS PŘEDMĚTŮ .....	51

## Seznam zkratk a značek

Application Programming Interface	API
Business Territory	BT
Comma-separated values	CSV
My Territory	MT
Unicorn College	UCL
Unicorn Enterprise System	UES
Unicorn Enterprise System Powered Company	UESPC
Unicorn Universe Business Modelling Language	UUBML
Unicorn Universe	UU
Unified Modelling Language	UML
Uniform Resource Identifier	URI

## Úvod

Tato bakalářská práce se zabývá rozšiřováním a implementací nových řešení v informačním systému Unicorn Universe (dále UU) pro společnost Unicorn College (dále UCL). UCL je soukromá vysoká škola, soustředící se na vzdělávání odborníků v oblasti informačních a komunikačních technologií, ekonomie a managementu a využívající ke svému provozu informační systém UU založený na platformě Unicorn Enterprise System (dále UES).

Informační systém Unicorn Universe je internetová služba, veřejně dostupná z umístění <https://unicornuniverse.eu/> a umožňující správu podniku a podnikových procesů, klientů, zakázek či úkolů a rovněž poskytující osobní digitální pracovní prostor, v níž lze ukládat neomezené množství dat, evidovat si veškeré smlouvy či si plánovat aktivity do diáře.

Cílem této bakalářské práce je implementace řešení na míru, optimalizace a automatizace dvou důležitých procesů v rámci subsystému Unicorn College.

Prvním důležitým bodem je publikace dosažených studijních výsledků pro studenty a účastníky kurzů školící společnosti Top Gun Academy, organizačně spadající pod Unicorn Learning Centre. Výstupem tohoto procesu je přehledná a ucelená tabulka s výsledky a statistickými údaji snadno dostupná pouze příslušnému studentovi.

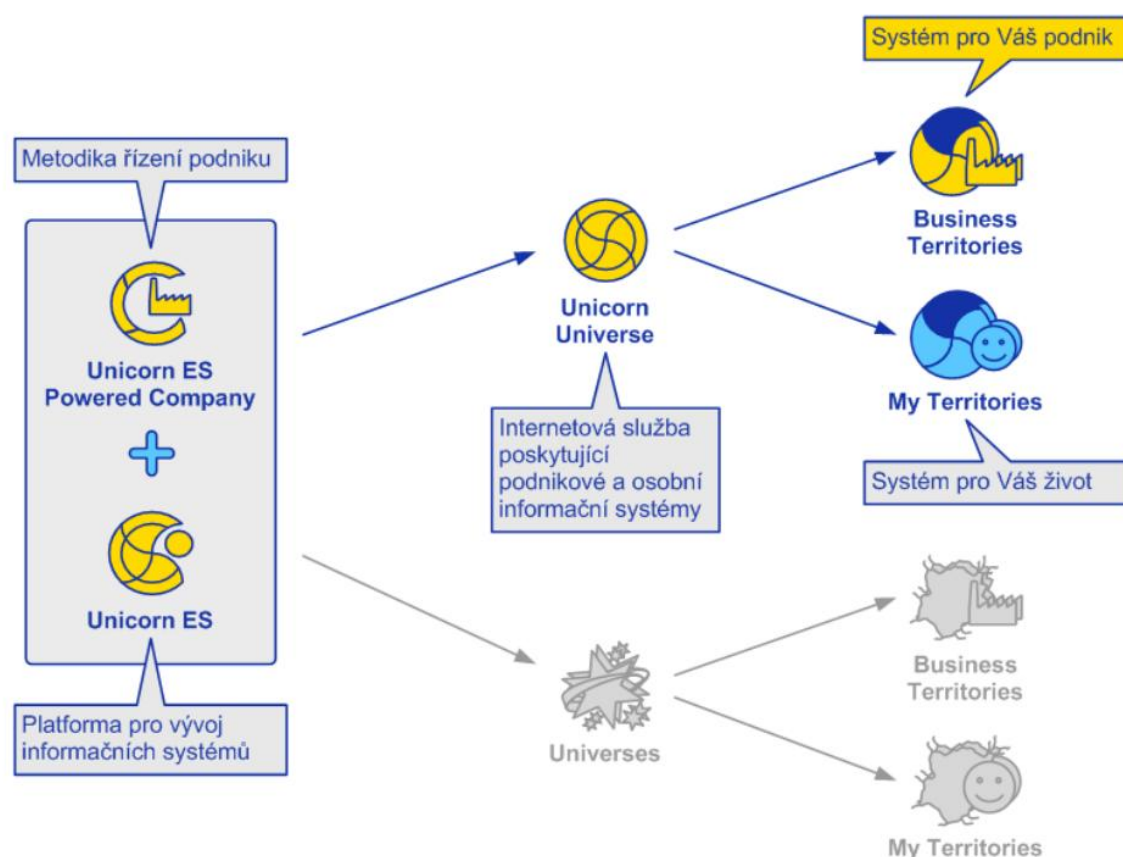
Druhým procesem je automatizace zápisu předmětů do akademického roku a jeho realizace v rámci Unicorn Universe, zahrnující hromadný zápis předmětů na základě zápisového listu a rovněž umožnění korekce zápisu se snahou minimalizovat zatížení studijního oddělení.

Proto je i nedílnou součástí bakalářské práce bližší seznámení s hlavními principy informačních systémů založených na platformě UES, s užitou metodologií Unicorn Enterprise System Powered Company (dále UESPC) a s modelovacím jazykem Unicorn Unified Business Modelling Language (dále UUBML), které jsou pro rozšíření informačního systému používány.

# 1 Informační systém Unicorn Universe

*Informační systém Unicorn Universe* je bezplatná internetová služba provozovaná na platformě UES, která je vyvíjena, rozšiřována a spravována stejnojmennou firmou v rámci holdingu Unicorn a. s. Díky nepřetržité dostupnosti systému mají uživatelé přístup ke svým informacím kdykoliv a odkudkoliv pouze pomocí internetového prohlížeče.

Základním principem platformy UES je umožnění vývoje a správy neomezeného množství informačních systémů, souhrnně označovaných jako *Universes*. Každý tento Universe je provozně, systémově i datově samostatnou jednotkou a umožňuje vyvíjet vlastní samostatné systémy. Hlavním cílem společnosti Unicorn Universe však není provoz co největšího počtu Universes, ale nabízení služeb v rámci primárního a nejdůležitějšího Universe, čímž je *Unicorn Universe*, viz obrázek 1.



Obrázek 1-1: Struktura platformy UES

Zdroj: Unicorn Enterprise System Powered Company, 2012

## 1.1 Sektory a teritoria

Architektura Unicorn Universe spočívá ve vertikálním rozdělení systému na hardwarově oddělené *sektory*. Každý sektor slouží jako kontejner pro teritoria a je provozován s vlastní databází využívající technologie Oracle a PostgreSQL. Tento systém sektorů zajišťuje zvýšenou stabilitu a zjednodušení nasazování nových verzí bez ohrožení přístupnosti. V případě výpadku jednoho sektoru tím pádem není nutný restart celého systému.

*Teritorium* je logicky odděleným datovým prostorem v rámci sektoru a je dvojího druhu, buď jde o pracovní Business Territory, nebo personální My Territory. V rámci jednoho Universe je možno provozovat neomezené množství teritorií. Teritorium je v hierarchii objektů nejvýše umístěným kontejnerem a platí, že všechny jiné objekty jsou součástí právě jednoho teritoria.

### 1.1.1 My Territory

Prvním druhem teritoria je *My Territory* (dále MT), zakládané uživateli automaticky po bezplatné registraci do systému. Služba MT je individuální, s důrazem na soukromí, bezpečí a nabízí digitální pracovní prostor k řízení každodenního života. MT může sloužit jako úložiště dat s jednoduchou katalogizací nebo ke správě pracovních povinností i volného času, k čemuž pomáhají funkčnosti diáře, úkolovníku či mailu. MT je ovšem úzce spojeno s pracovními pozicemi (viz. *Role*) v Business Teritoriích (dále BT). Pokud UU využíváte i ke své práci a máte pracovní role v BT, tak se Vám veškeré úkony zadávané na jakoukoliv z vašich pracovních rolí okamžitě promítají do vašeho diáře i úkolovníku.

Přístup do MT zajišťuje personální role, která je každému uživateli zřizována právě jedna. Tím pádem platí, že počet personálních rolí zhruba odpovídá počtu registrovaných uživatelů, ačkoli se samozřejmě jedna osoba může zaregistrovat i vícekrát.

### **1.1.2 Business Territory**

*Business Territory* je naopak digitální pracovní prostor určený k řízení firmy a všech firemních procesů, například řízení zakázek a úkolů, správy zaměstnanců, klientů, majetku, know-how i řízení organizační struktury.

Každé BT je tedy komplexním informačním systémem. Ačkoliv je každé teritorium založeno na metodice UESPC, tuto metodiku je v rámci BT možné upravovat k vlastním potřebám, což umožňuje UU dodávat velice variabilní a unikátní řešení na míru, vedoucí k rozmanité základně společností využívajících UU, například vysoká škola Unicorn College, Český rybářský svaz, Kühnův dětský sbor nebo i Unicorn a. s.

Všechna BT mohou být podrobněji strukturalizována do podřízených celků, k čemuž nám slouží organizační jednotka popsaná v následující kapitole.

## **1.2 Organizační jednotka**

*Organizační jednotka* je prvním důležitým nástrojem jak strukturalizovat teritorium či podnik dle určité hierarchie. Hlavní vlastností organizační jednotky je diferenciací práv uživatelů ke čtení a editaci objektů. Aby se mohl uživatel „pohybovat“ po organizační jednotce, musí být obsazen v roli, která se dané organizační jednotky přímo týká (viz následující kapitola). Daný uživatel však přesto nemá přístup ani do podřízených, ani do nadřízených organizačních jednotek, pokud se nenachází v roli, jež by mu tato práva poskytovala.

## **1.3 Role**

*Role* je základní reprezentací reálné osoby v systému a dělí se na dva základní druhy, na role přístupové a pracovní.



### 1.3.1 Přístupové role

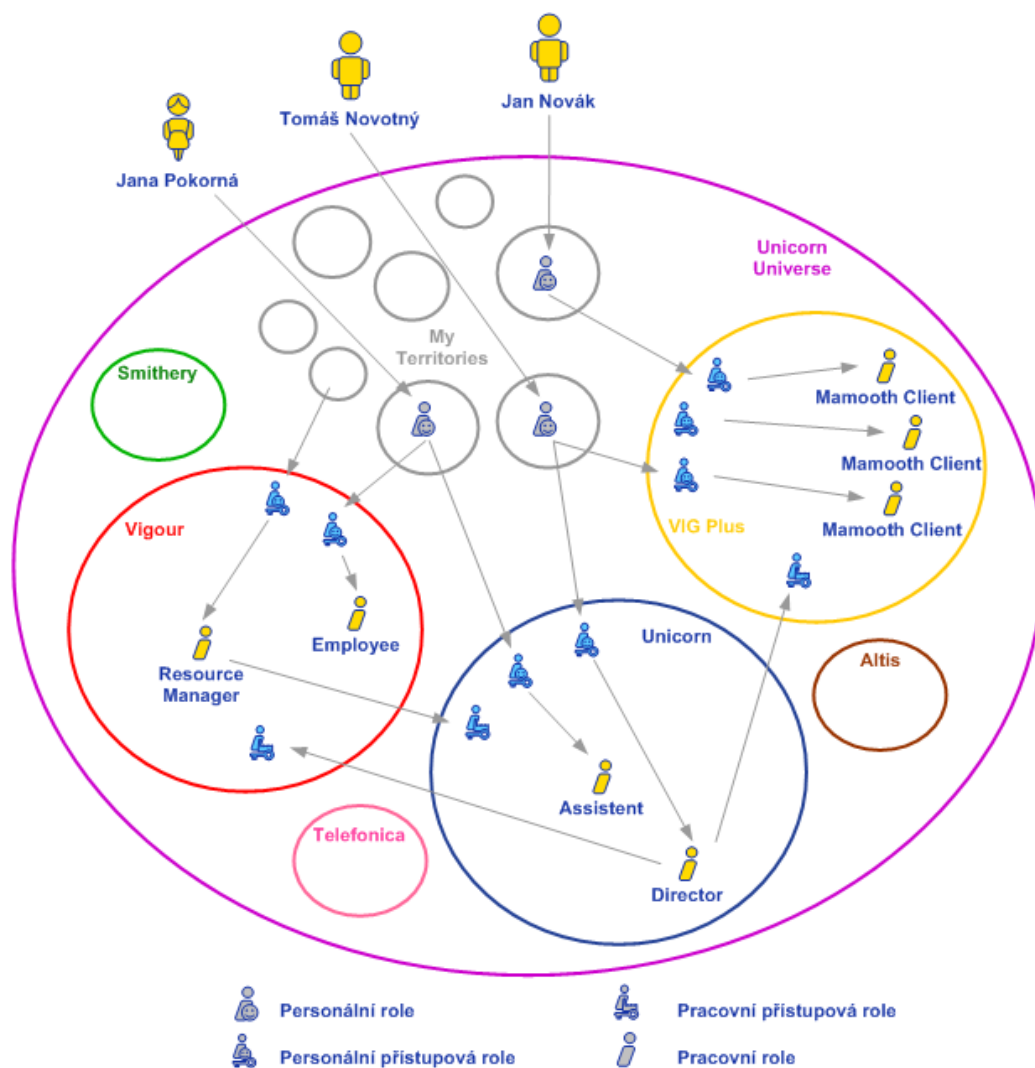
*Přístupové role* slouží především ke správě přístupu uživatele do teritoria a určují mu základní podmínky k práci a pohybu. V současné době existují tři druhy přístupových rolí.

Prvním typem je role osobní (Personal Role). Tato role je absolutním základem pro užívání systému, neboť zprostředkovává přístup do MT a je tedy reprezentací existující osoby. Tuto roli má každý uživatel pouze jednu a je spravována pouze jejím originálním vlastníkem. Jedinečnou identifikaci osobní role zajišťuje UID – Universe ID, což je systémem generovaná číselná kombinace. Příkladem UID je 5-8385-1.

Dalším typem je role *osobní přístupová* (Personal Access Role), umožňující pohyb a práci v BT. Tuto roli zakládá pouze správce přístupových rolí (což je mimo jiné speciální pracovní role) každému uživateli zvlášť a pouze jednu na celé teritorium. Tyto role často slouží k reprezentaci vztahu osoba – společnost (například zaměstnanec – firma nebo klient – obchod).

Posledním typem jsou *pracovní přístupové role* (Business Access Role), které mají obdobný účel jako Personal Access Role. Základní rozdíl však je, že Business Access Role se používají pro spolupráci mezi dvěma BT, například klient - dodavatel. Hlavní předností je nezávislost na konkrétní osobě. Příklad: Business Access Role je zřízena do teritoria ABC pro ředitele společnosti XYZ, pana Jana Nováka. Pokud se však ředitelem společnosti XYZ stane Eliška Pokorná, převezme tak bez problémů aktivity týkající se firmy ABC.

Na příkladu na obrázku 1-2 je možné lépe pochopit problematiku přístupových rolí. Uživatel Tomáš Novotný vlastní teritorium s personální rolí a má zařízenou osobní přístupovou roli do teritoria Unicorn, v níž pracuje na pozici ředitele. Tato role ředitele má zřízenou pracovní přístupovou roli do teritorií VIG Plus a Vigour. Pokud by se ředitelem stala Jana Pokorná, získá tím navíc přístup do teritoria VIG Plus a uživateli Tomáši Novotnému zůstane přístup pouze do teritorií Unicorn a VIG Plus z důvodu existujících osobních přístupových rolí.



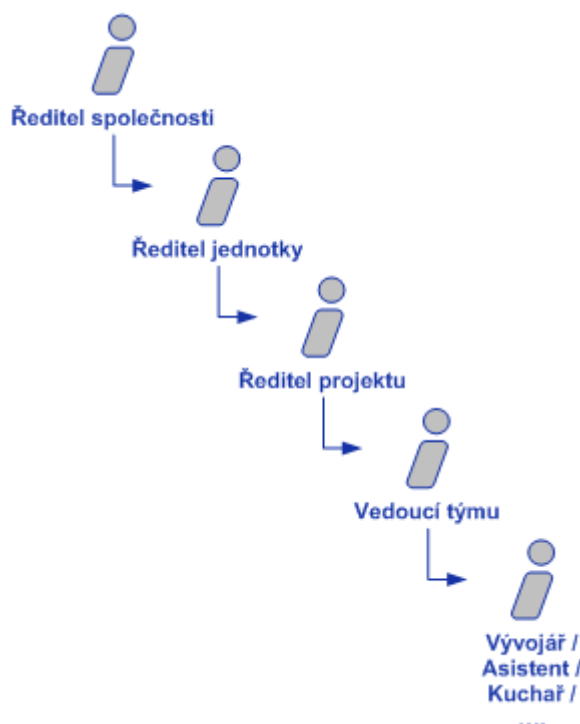
Obrázek 1-2: Příklad teritorií a rolí

Zdroj: Unicorn Enterprise System Powered Company, 2012

### 1.3.2 Pracovní role

*Pracovní role* je systémovou reprezentací pracovní pozice ve firmě, nezávisle na osobě, která danou pozici vykonává, ačkoli hlavní obsazení konkrétní osobou je u dané pracovní role vždy viditelné.

Pracovní role je důležitým nástrojem k vytváření organizační struktury podniku a BT. Za každou roli v systému je kompetentní jiná role, čímž je modelován vztah nadřízenosti a podřízenosti (viz. Obrázek 1-3: Příklad kompetence rolí).



Obrázek 1-3: Příklad kompetence rolí

Zdroj: Unicorn Enterprise System Powered Company, 2012

Důležitý případ užití týkající se role je *obsazování*. Do každé role (pracovní i přístupové) je možné obsadit jinou roli, což v praxi znamená, že je mezi rolemi i jiný vztah než nadřízenost a podřízenost. Příkladem budiž ředitel + asistentka, přičemž do pracovní role ředitele je obsazena role asistentky ředitele, která tímto způsobem získává téměř stejná přístupová práva k objektům v teritoriu jako samotný ředitel.

Na tomto místě je nutné zmínit, že existují tři úrovně obsazení role, každá se svými specifikacemi a standardními použitími.

První úrovní obsazení je obsazení *výkonné*. Veškeré úkoly směřované na tuto roli jsou přímo propagovány všem rolím obsazeným výkonně, znamenající, že právě ony by měly být přímými řešiteli daných aktivit. Z tohoto důvodu je doporučeno mít výkonně obsazeno právě jednu roli.

Další možností je obsazení *asistenční*, jehož hlavní rozdíl spočívá v tom, že aktivity (úkoly, schůzky, otázky) nejsou zobrazovány rolím obsazeným asistenčně. Primárním využitím je tedy přenesení práv dané role na větší množství uživatelů. Předem zmiňovaný

příklad s ředitelem a jeho asistentkou má tedy tu výhodu, že asistentka může snadno spravovat artefakty (viz kapitola 1.3) týkající se role ředitele.

Poslední možností, jak vzájemně obsazovat role, je *host*, nesoucí standardně nejnižší práva přístupu k objektům. Tato úroveň obsazení je používána především v případě snahy přidělit práva pouze ke čtení. Role obsazené v úrovni *host* mohou standardně pouze prohlížet obsah, případně do obsahu přidávat komentáře (typ agregovaného objektu k artefaktu, viz kapitola 1.4.1).

Pracovní role je tedy možné vzájemně obsazovat, dalším rozlišením a způsobem jak strukturálně uspořádat teritorium jsou různé druhy rolí, a to běžná pracovní role pro jednotlivce, skupinová role a *power role*.

Do všech 3 druhů rolí je možné obsazovat na jakoukoliv úroveň, hlavním rozdílem je ale delegování aktivit náležících rolí a způsob, jakými jsou používány.

Mějme roli jednotlivce s názvem IT Projectant 2, v níž jsou obsazeny 3 další role, všechny výkonně a roli skupinovou s názvem IT Projectants, v níž je výkonně obsazeno 5 pracovních rolí. Nyní roli IT Projectant 2 a IT Projectants zadáme úkol „Donést čaj“. V případě role IT Projectant 2 se úkolu chopí jedna z 3 obsazených rolí, úkol splní a zadavateli úkolu bude donesen jeden čaj. V druhém případě však dojde k delegaci úkolu na všechny obsazené role a každá z nich se bude snažit daný úkol splnit. V optimálním případě tak dojde k donesení 5 čajů.

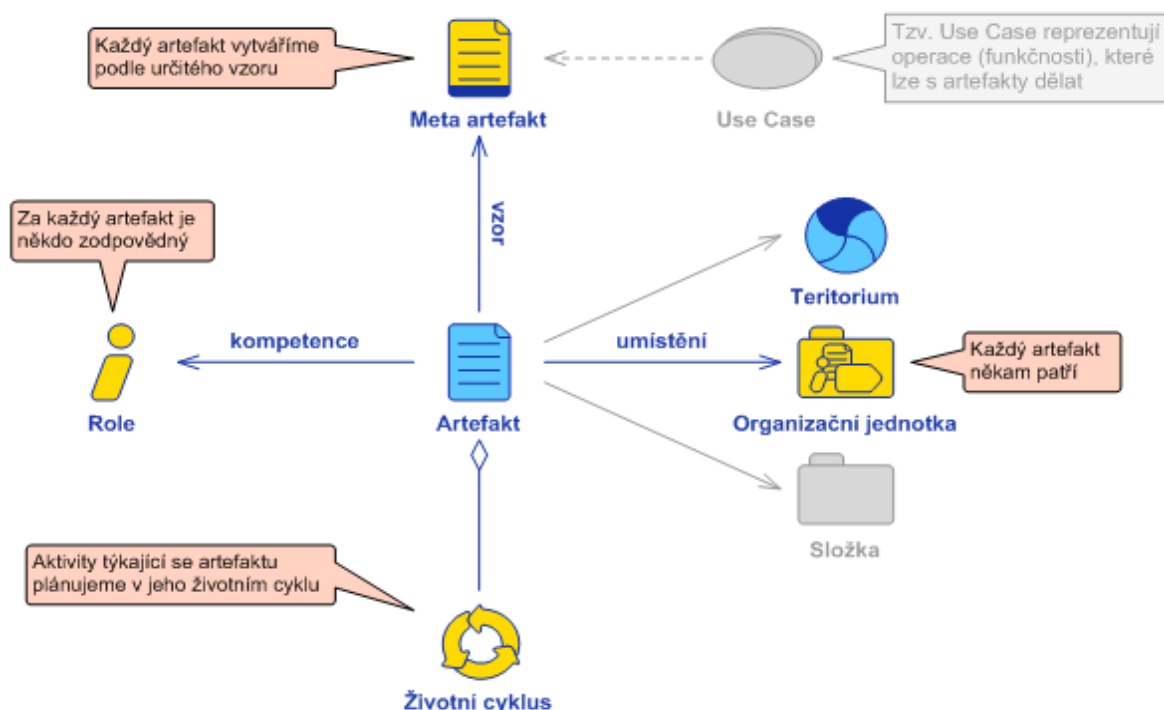
Obdobným způsobem se chovají i *Power role*, nicméně ty slouží k naprosto jinému účelu. Do *Power rolí* by měli být obsazovány pouze role zodpovědné za celý proces, tedy top management, případně role kompetentní za danou organizační jednotku. Další charakteristikou je implicitní oprávnění přístupu do všech podřízených organizačních jednotek pro obsazené role.

## 1.4 Artefakt

*Artefakt* je základním nositelem informace v Unicorn Universe a jde o elementární stavební kámen systému, přičemž platí, že každý objekt je artefaktem nebo jeho agregovaným objektem a zároveň každý případ užití se přímo či nepřímo alespoň jednoho artefaktu týká. V hierarchickém uspořádání objektů je artefakt níže než teritorium (které je také samo o sobě zvláštní třídou artefaktu).

Artefakt si lze představit jako chytrý dokument, umožňující informace nejen evidovat, ale i řídit. Dobrým příkladem je Karta studenta v Unicorn College teritoriu. Z obsahové části je možno evidovat základní informace o studentovi, průběh studia, zapsané předměty a relevantní studijní výsledky. Z řídicího pohledu je možné plánovat pohovory, zkoušky, spravovat aktuální stav studenta či zadávat domácí úkoly.

Artefakt je vysoce variabilní a diferenciovaný do několika tříd. Kromě obyčejného artefaktu, který má právo zakládat každý uživatel, je artefaktem například i role, složka, organizační jednotka, teritorium atd.



Obrázek 1-4: Hlavní parametry artefaktu

Zdroj: Unicorn Enterprise System Powered Company, 2012

Každý artefakt má několik základních parametrů, jak je znázorněno na obrázku 1-4. Každý artefakt má minimálně tuto sadu identifikátorů, různé podtřídy artefaktů však mohou mít vlastní specifické vlastnosti.

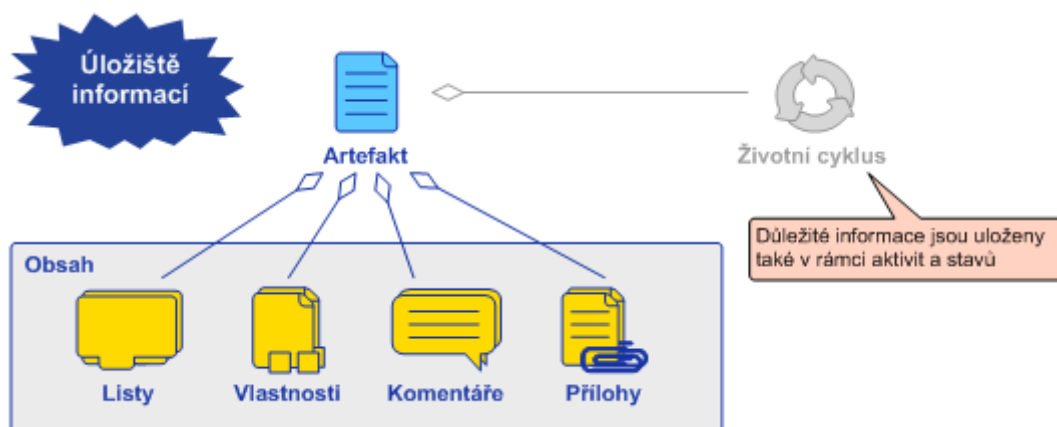
- Identifikace artefaktu je zajištěna názvem a kódem. Název artefaktu věcně pojmenovává účel, ke kterému artefakt existuje, přičemž názvy mohou být duplicitní. Naopak kód artefaktu slouží jako kandidátní klíč, jehož unikátnost musí být zachována v rámci teritoria.
- Za každý artefakt musí být kompetentní právě jedna role z důvodu jednoznačného určení osoby zodpovědné za informace relevantní k artefaktu. Tato role má zpravidla nejširší výběr autorizovaných funkcí.
- Strukturalizace na složky, organizační jednotky a teritoria umožňuje přehledně katalogizovat artefakty z hlediska umístění. Z laického pohledu na věc je možné umístění chápat podobně jako systém složek na většině operačních systémů, nicméně artefakty je možné adresovat přímo přes jejich primární či kandidátní klíč.
- Každý artefakt (karta studenta, domácí úkol, technická dokumentace, pozvánka na meeting) je zakládán dle konkrétní šablony (dle meta artefaktu), který definuje přednastavený obsah, životní cyklus, vlastnosti, zobrazení a rovněž i třídu. V objektovém pohledu je možno meta artefakt definovat jako rodiče a artefakt jako potomka.

#### **1.4.1 Agregované objekty**

Artefakty sami o sobě jsou nositeli přidružených objektů umožňujících podrobněji strukturalizovat požadovanou informační hodnotu artefaktu.

Tyto *agregované objekty* jsou vždy vázány právě na jeden artefakt, který se stává jejich umístěním. K dalším parametrům každého objektu patří název a kód, který musí být zachován unikátní v rámci celého artefaktu, není tedy možné mít například list a vlastnost se stejným kódem.

Typy agregovaných objektů jsou vidět na obrázku 1-5 níže:



Obrázek 1-5: Typy agregovaných objektů

Zdroj: Unicorn Enterprise System Powered Company, 2012

*Listy*, jak již název napovídá, jsou svojí funkcí podobné běžným textovým editorům. Slouží především k uchování a sdílení textových a obrazových informací středního až většího rozměru, u kterých je nutno zachovat jejich vizuální formu. Obsah listu je možné upravovat editorem typu WYSIWYG (What You See Is What You Get) integrovaným přímo do Unicorn Universe. Architektura a koncepce listu je řešena pomocí značkového jazyka XML, který umožňuje obsah listu přesně a strukturovaně upravovat i pomocí skriptů a servisních, automatizovaných zásahů.

*Vlastnost* (neplést s parametrem artefaktu či objektu) je naopak nositelem přesně stanoveného druhu informace. Každá vlastnost má svůj specifický datový typ, který po založení vlastnosti není možné změnit (například text, číslo, datum, obrázek, binární vlastnost nebo reference neboli odkaz na objekt). Primární případ užití týkající se vlastností z uživatelského hlediska je tvorba formulářových polí, umožňujících snadno naplňovat vlastnosti hodnotami. Tyto hodnoty je pak snadné automaticky zpracovat.

*Příloha* je dalším druhem agregovaného objektu a svým použitím připomíná přílohu e-mailu, umožňující k artefaktu nahrát jakýkoliv soubor.

Zvláštním druhem agregovaného objektu jsou *komentáře*. Zpravidla je umožněno přidávat komentáře všem rolím, které mají právo čtení popisu artefaktu. Slouží především k vyjádření vlastních poznatků, připomínek či dotazů k obsahu artefaktu. Ke komentování slouží místa určená uživatelem v popisu artefaktu, takzvané komentační body.

Velice důležitou součástí artefaktu umožňující dynamicky řídit informaci je životní cyklus, jemuž je věnována následující kapitola 1.4.2.

### 1.4.2 Životní cyklus

*Životní cyklus* je primární způsob jak dynamicky řídit a spravovat informace. Životní cyklus je možné chápat, jak již název napovídá, jako soubor stavů a činností týkajících se artefaktu v časovém sledu od založení až po uzavření artefaktu. Elementárními prvky životního cyklu jsou možnosti regulovat stav artefaktu a plánovat aktivity.

Každá informace se v čase vyvíjí, prochází různými stavy, může se změnit, či zaniknout. Reprezentací časového vývoje je stav artefaktu, jakožto nedílná součást životního cyklu. Stav artefaktu odráží aktuální stav informací, příkladem budiž softwarový projekt na obrázku Obrázek 1-6: .



Obrázek 1-6: Stavy artefaktu Softwarový projekt

Zdroj: vlastní

Projekt je třeba nejprve založit, vytvořit zadání, přijmout, revidovat zadání, implementovat, a tak dále, dokud se nedostane do finálního stavu – Nasazeno. Každý tento stav je možno na artefaktu reprezentovat a je již na první pohled vidět, v jaké fázi se projekt nachází. Samozřejmě je možné nastavit i alternativní stavy, které nesledují cestu tohoto „happy-day“ scénáře, jako je například stav Zrušeno, Pozastaveno nebo Nepřijmuto.

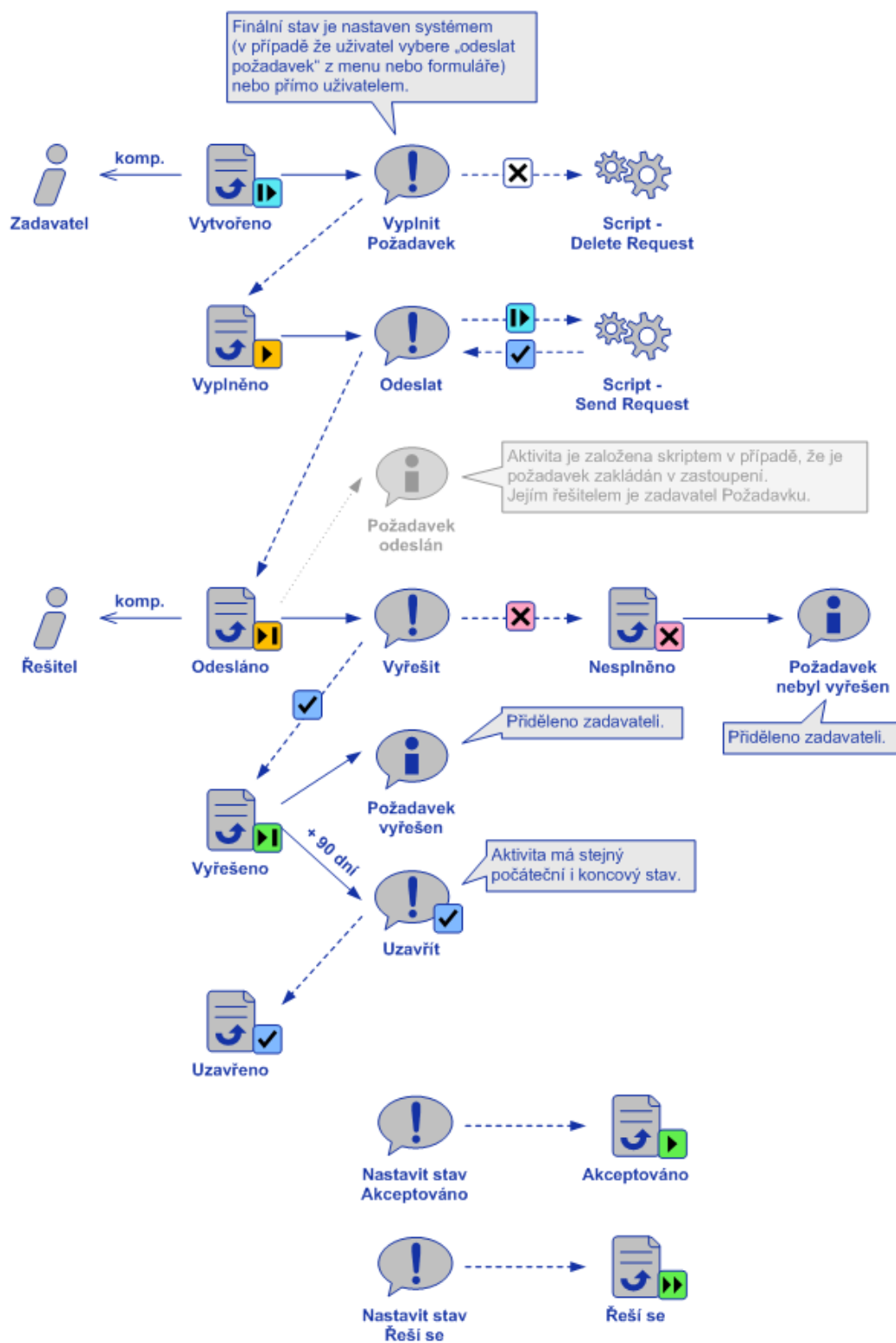


*Aktivita*, jak je chápána z pohledu Unicorn Universe, je konkrétní činnost nebo úkol, jež by měla být provedena v souvislosti s účelem artefaktu v časovém horizontu vymezeném zadáním aktivity. Tím je zajištěno dynamické řízení stavu informace, evidence veškerých změn artefaktu a možnost zadávat a sledovat vývoj úkolů v čase.

Na aktivitách je z tohoto důvodu také možné určit stav, který vyjadřuje míru rozpracovanosti daného úkolu. Každá aktivita má svého zadavatele a jednoho nebo více řešitelů a každá tato strana má právo nastavovat určité stavy aktivit.

Aktivita je ovšem obecný pojem a dělí se na několik typů. Mezi generické aktivity, možné založit na jakémkoliv artefaktu, patří aktivita typu Úkol, Zpráva, Schůzka, Dotaz nebo Rozhodnutí. Je však možné si vytvořit i vlastní vzor aktivity, více viz kapitola 1.6.

Životní cyklus je tedy důležitým nástrojem k dynamickému rozvíjení artefaktu. Například na artefaktu třídy požadavek je implicitně nastaven poměrně složitý životní cyklus reagující na různé standardní i alternativní scénáře. Požadavek je druh artefaktu zakládáný přímo nad jiným artefaktem určitou rolí (tyto informace si požadavek ukládá do vlastních parametrů specifických tříd artefaktu Požadavek – jako parametry *Předmětný artefakt* a *Role zadavatele*). Slouží především k zadávání úkolů, u nichž nás nezajímá průběh řešení, ale pouze finální výsledek (například požadavek na Helpdesk, požadavek na obsazení apod.). Příklad životního cyklu je k vidění na obrázku Obrázek 1-7: Životní cyklus požadavku.



Obrázek 1-7: Životní cyklus požadavku

Zdroj: Unicorn Enterprise System Powered Company, 2012

## 1.5 Unicorn ES Uniform Resource Identifier

*Uniform Resource Identifier* (dále URI) je jednoznačný identifikátor zdroje informace ve formě textového řetězce s předem určenou strukturou jak stanovil Berners-Lee, Fielding a Masinter, 2005 (viz vzorec 1).

*schéma: hierarchická část? dotaz&fragment*

Vzorec 1: Obecná struktura URI

Zdroj: Berners-Lee, 2005

Schéma určuje hierarchickou strukturu a povolenou syntaxi pro zbývající část URI. Samotné schéma je textový řetězec, který se může skládat pouze z alfanumerických znaků, případně znaku plusu, mínusu nebo tečky. V rámci systému Unicorn Universe je jednotně zavedeno schéma „ues“.

Hierarchická část nám vypovídá o uspořádání objektů. Jak již bylo zmíněno dříve, v Unicorn Universe systému je hierarchicky nejvýše postaveným objektem teritorium, níže je artefakt a posléze jeho agregované objekty. Každý tento objekt má v Unicorn ES URI (dále UESURI) dva klíčové identifikátory, tím jest kód a Object ID (dále OID). Kód je proměnlivý a uživatelsky změnitelný, je však nutno zachovat jeho unikátnost v hierarchicky nadřazeném umístění. OID je naproti tomu systémem generovaný číselný identifikátor, který není možné po založení měnit, pouze číst.

Vzhledem k jednoznačnosti obou klíčů je tedy možné adresovat objekty více způsoby a více formami UESURI. První z nich je adresace pouze pomocí kódů, ve formátu, jež je vidět ve vzorci 2.

*ues: kód teritoria: kód artefaktu: kód objektu*

Vzorec 2: Prostá forma UESURI

Zdroj: vlastní

Z výše uvedeného vzorce je tedy možné rozpoznat, že jednotlivé hierarchické prvky jsou oddělovány dvojtečkou. Kompletní, normalizovaná forma UESURI však počítá jak s kódy, tak s OID objektů (viz vzorec 3). V neposlední řadě také kanonická forma UESURI, která je vhodná především pro optimalizaci rychlosti a datové zátěže, neboť je složena pouze z primárních klíčů.

*ues: kód teritoria[OID teritoria]: kód artefaktu[OID artefaktu]: kód objektu[OID objektu]*

Vzorec 3: Normalizovaná forma UESURI

Zdroj: vlastní

Poslední, volitelnou částí UESURI (a obecně i URI) je dotaz a fragment, v terminologii Unicorn Universe je dotazem myšlen Use Case. Případem užití může být například vytvoření aktivity, požadavku, nahrání přílohy. Fragment může být spojen s popisem artefaktu, kdy odpovídá XML značce nebo jejímu atributu, případně může jít o parametry případu užití (například název zakládané aktivity).

## 1.6 Metodický artefakt

Každý artefakt v systému je založen dle určité třídy, kterou mu určuje rodič – *metodický artefakt* (dále meta artefakt). Meta artefakt slouží jako šablona, podle které je možné vytvářet identické potomky.

Dle standardů Unicorn ES Powered Company jsou meta artefakty umístěny do Meta modelu odpovídajícím určitému procesu (například Production, Sales, Management apod.) a jsou spravovány rolí metodické autority kompetentní za daný meta model.

Při zakládání artefaktu si vždy volíme z meta artefaktu, dle kterého vytváříme potomka. Tohoto potomka je následně možné libovolně měnit, aniž bychom prováděli jakoukoliv změnu v meta artefaktu.

Meta artefakt nám umožňuje nastavit vzory pro všechny agregované objekty i pro základní parametry artefaktů, jako je výchozí název nebo kód. Listy jsou vždy zakládány dle osnovy, vlastnosti dle vzorů vlastností a životní cyklus je generován dle vzorového životního cyklu. Tímto způsobem jsou řešeny vlastní metodiky jednotlivých teritorií, například v meta modelu Unicorn College se nachází meta artefakt Karta studenta, Elektronický index, Výuková aktivita, Portál předmětu, Karta zaměstnance apod. V organizační jednotce se studenty tak můžeme mít veškeré informace přesně a jasně katalogizované v artefaktech, dle předem dané struktury, vzhledu a s možností tyto informace řídit jedním, uceleným způsobem.

## **1.7 Autorizační pravidla**

V systému Unicorn Universe je kladen veliký důraz na bezpečí a naprosté soukromí všech uložených dat. Vzhledem k tomu, že jde o veřejně přístupnou službu, je nutné mít neproniknutelné zabezpečení nejen před vnějšími útoky, ale i před samotnými uživateli, čemuž je tato kapitola věnována. Kvůli tomu byl vyvinut složitý systém *autorizačních pravidel* limitující práva pro čtení a editaci artefaktů pro specifické skupiny uživatelů.

Nejdůležitějším faktorem je rozhodně systém rolí, který zajišťuje přesné určení, kteří uživatelé mají přístup do daného teritoria či organizační jednotky. Základní omezení přístupu do teritoria nám tedy zajišťuje přístupová role, zatímco přístup do organizační jednotky (která je „menším“ objektem než teritorium) je specifikována rolí pracovní (viz kapitola 1.3).

Nicméně toto není jediný způsob jak zajistit přístupnost informací, neboť by bylo systémově náročné a odporující metodice Unicorn ES Powered Company vytvářet pracovní roli ke každému artefaktu, jenž se nenachází v organizační jednotce uživatele. Pracovní role má přímo odrážet pozici ve firmě. Z tohoto důvodu je možné přístupová práva k objektům více separovat přes různé stupně prověření a utajení a také práva přidělovat implicitně a explicitně.

### 1.7.1 Stupeň prověření a utajení

*Stupeň prověření* je jedna ze základních vlastností přístupové role do teritoria, která určuje bezpečnostní prověrku udělenou správcem teritoria, zatímco *stupeň utajení* je vlastnost nastavitelná na každém artefaktu. Z pohledu autorizačních algoritmů jde o základní prerekvizitu přístupu k jakémukoliv objektu, neboť tyto dvě hodnoty jsou porovnávány jako první.

Pokud uživatel ale nemá dostatečný stupeň prověření na určitý artefakt (například se stupněm prověření Běžné na přístupové roli a stupněm utajení Důvěrné na artefaktu), nemá uživatel možnost tento artefakt si nijak zobrazit, ani přes vyhledávací funkce systému. Toto je rovněž základním rozdílem oproti dalším krokům autorizace.

### 1.7.2 Implicitní a explicitní práva

Jak již název napovídá, existují dva způsoby standardního přidělování práv k objektům. Každá role je vůči všem artefaktům v jasně daném vztahu, například se může nacházet ve stejné organizační jednotce, ve stejném teritoriu nebo je roli na daném artefaktu delegován úkol. Tato forma vazby je vyjadřována druhem implicitní skupiny, k níž je možné na meta artefaktu specifikovat práva ve všech přístupových bodech a možných případech užití. *Implicitní práva* jsou tedy způsobem, jak omezit uživatele na všech artefaktech založených dle jednoho metodického artefaktu.

Naopak *explicitní práva* jsou specifická jednomu konkrétnímu artefaktu a je možné jimi přidělovat práva zvolené roli. Tím pádem je trochu poodkryt autorizační algoritmus. Nejprve je ověřována úroveň prověření a utajení, jenž je naprostým předpokladem přístupu. Poté jsou kontrolovány práva explicitní. Pokud i přes to dojde k autorizační výjimce, následuje kontrola práv implicitních. V případě neúspěchu je pak uživatel varován chybovou hláškou na nedostatečné oprávnění.

## 2 Nástroje Unicorn

Lepší postavení firmy na trhu a větší konkurenceschopnost je možno zajistit několika způsoby, například luxusním výrobkem, jenž je jinde nedostupný, lepší cenou nebo kvalitnějším servisem. Firma Unicorn a.s. navrhla a na praktické úrovni i používá několik základních pravidel, které ji vedou k úspěchu na poli vývoje softwaru. Tento přístup se nazývá *Unicorn Approach*:

*„Unicorn Approach (Přístup Unicorn) je principiální přístup k podniku a podnikání, ve kterém z vámi zvoleného portfolia produktů a služeb a s ohledem na vámi preferovanou dělbu práce, preferované technologie a odhadovaná rizika, odvodíte organizační strukturu, popíšete a předepíšete pracovní postupy jednotlivých podnikových procesů pro všechny zúčastněné role, zdůrazníte klíčové produkty a meziprodukty, pro které rovněž předepíšete strukturu informací a uvedete tento mechanismus v život. Váš podnik se pak stane dobrým systémem, který více méně předvídatelně plní účel, pro který byl zřízen.“* Kovář Vladimír (2011, s.16)

K dosažení těchto stanovených cílů slouží 2 hlavní nástroje. Prvním z nich je vlastní metodika Unicorn Enterprise System Powered Company (dále UESPC), jenž slouží jako návod pro řízení podniku a na níž je vystavěn systém Unicorn Universe. Tím dalším, neméně důležitým, je modelovací jazyk Unicorn Unified Business Modelling Language (dále UUBML), sloužící především k modelování a vizuálním manifestaci schémat, návrhů a modelů.

### 2.1 Unicorn Enterprise System Powered Company

*UESPC* je metodika vyvinutá pro přesné řízení podniku s jasnou organizační strukturou a vychází z dlouholetých zkušeností Vladimíra Kováře při vedení společnosti Unicorn, ve které je rovněž v plné míře aplikována.

Tato metodika je integrována jako elementární myšlenka a stavební kámen UU, nicméně ji lze aplikovat i zcela nezávisle na této platformě. Bohužel se dle Vladimíra Kováře (2011, s.17) na současném trhu informačních systémů nenachází žádný natolik automatizovaný software, jenž by dokázal postihnout klíčové procesy metodiky UESPC.

Hlavním subjektem UESPC je optimalizace řízení podniku, na nějž je nahlíženo ze dvou úhlů, ze statického a dynamického. Statický pohled modeluje především organizační strukturu, vytvářené produkty či služby a vztahy mezi nimi. Ze systémového pohledu jde tedy o artefakty, jenž jsou strukturalizovány do teritorií, organizačních jednotek či složek.

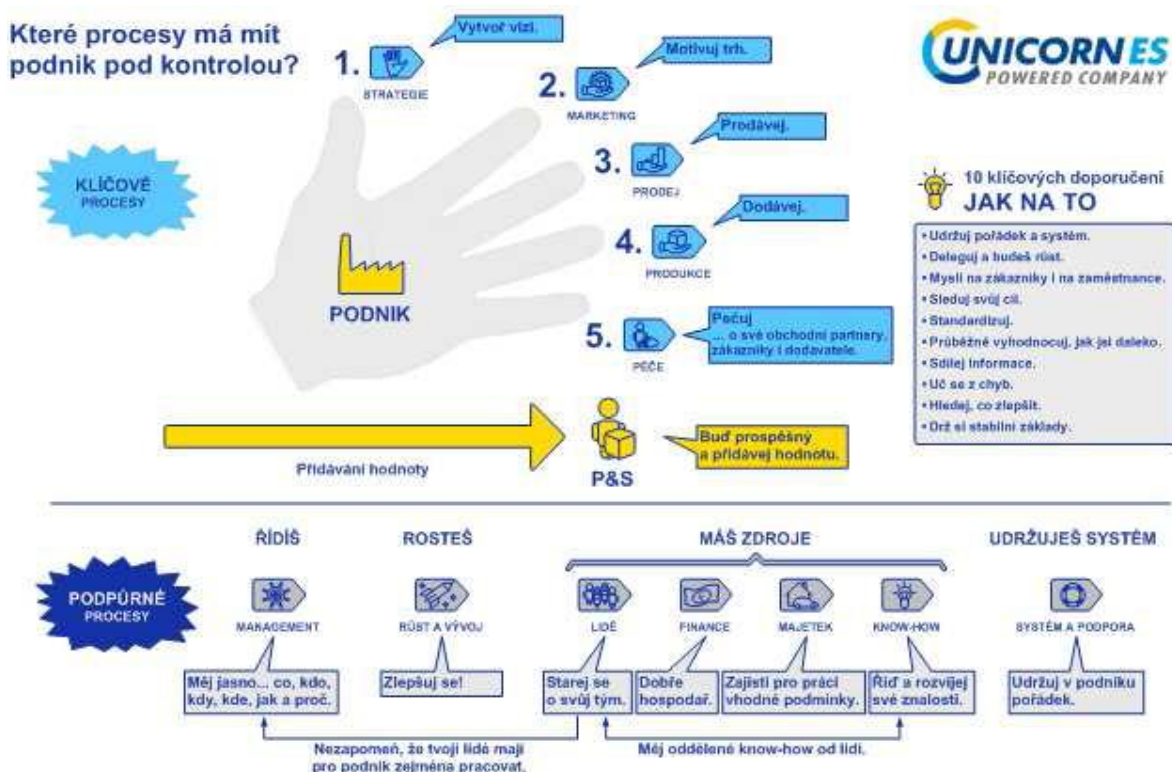
Dynamický pohled na podnik naopak zachycuje tok informací v čase, jehož základní jednotkou je tedy činnost neboli proces. Procesem rozumíme posloupnost akcí vedoucích k jasně definovanému cíli, v systému je pak proces dekomponován na své dílčí elementy, které jsou v konečném důsledku reprezentovány aktivitami v životním cyklu artefaktů.

UESPC definuje 12 procesů nezbytných k úspěšně fungující firmě a pro každou tuto oblast definuje základní myšlenky, postupy a metodické předpisy (v rámci systému tedy metodické artefakty a doporučenou organizační strukturu). Důsledná tvorba a správa těchto oblastí je ve společnosti Unicorn zajišťována konzultanty a metodiky.

- **Klíčové procesy:** Strategie, Marketing, Prodej, Produkce, Péče
- **Podpůrné procesy:** Management, Růst a vývoj, Lidé, Finance, Majetek, Know-how, Systém a podpora

Tyto klíčové a podpůrné procesy jsou popsány s komentáři na obrázku Obrázek 2-1: 12 procesů metodiky UESPC.





Obrázek 2-1: 12 procesů metodiky UESPC

Zdroj: Kovář Vladimír, 2011, str.22

Metodika Unicorn ES Powered Company byla inspirována z několika zdrojů a vznikla tedy výběrem a úpravou myšlenek jiných metodik, jež byly opracovány zkušenostmi a optimalizací při aplikaci na společnost Unicorn. Mezi důležité zdroje těchto myšlenek patří metodiky Rational Unified Process (RUP), jež přispěla iteračním přístupem a rozdělením vývoje softwaru do 4 fází. Dále také metodika Projects In a Controlled Environment (PRINCE2) a v neposlední řadě také Information Technology Infrastructure Library (ITIL), zajímavá především ze znalostního managementu.

## 2.2 Unicorn Unified Business Modelling Language

*Unicorn Unified Business Modelling Language* (dále UUBML) je jednoduše pochopitelným, kompaktním modelovacím jazykem sloužící ke komunikaci a ke znázornění objektů a vztahů mezi nimi.

Syntaktická i sémantická kostra tohoto jazyka je založena na standardu modelování v informačním systému, tedy na Unified Modelling Language (UML), nicméně je kladen veliký důraz na intuitivnost a srozumitelnost, jež UML nenabízí. Pro představu, všechna schémata a obrázky uvedené v této bakalářské práci jsou kresleny pomocí jazyku UUBML.

Nicméně regulující pravidla samozřejmě existují a jsou dodržovány i v tomto syntakticky volném jazyce. Distribuce a dohled nad těmito pravidly je řízen interně ve společnosti Unicorn, která svým zaměstnancům umožňuje si stáhnout a instalovat jazyk UUBML jako vzorník ikon do modelovacího programu Microsoft Visio.

Tyto ikony jsou ale rovněž snadno pochopitelné a využívají buď podobnosti svému vzoru, nebo jsou zvětšenou formou ikon denně viditelných v systému Unicorn Universe, viz Obrázek 2-2: . V této ilustraci jsou ikonky rovněž barevně odlišeny, přičemž každá barva reprezentuje rozdílnou důležitost objektu ve schématu, kdy šedé objekty jsou na normální úrovni důležitosti, modré na střední důležitosti a žluté jsou objekty nejvyšší priority.



Obrázek 2-2: Ikony UUBML

Zdroj: vlastní

Dalším důležitým standardem je dodržování barevného vzoru stavů artefaktů a aktivit, k čemuž slouží předloha tzv. „*Semafor stavů*“, kdy každá barva reprezentuje určitý typ stavu, jako je počáteční, aktivní, problémový, apod. Ilustrace tohoto je možné vidět a blíže si jí představit na Obrázek 1-6: .

Neméně důležitým prvkem jazyka je i možnost vyjádření vztahů mezi objekty. K této indikaci slouží konektory, jejichž vzhled je k vidění na Obrázek 2-3: Relační konektory UUBML. Každý konektor reprezentuje odlišný způsob relace mezi objekty:

- **Normální:** značí proces nebo obecné vztahy mezi objekty (například student chodící na přednášku)
- **Agregační:** vyznačuje, že je jeden objekt součástí druhého (příloha, listy, životní cyklus jsou součástí artefaktu)
- **Asociační:** jeden objekt je odkazován druhým (například je v popisu artefaktu uveden odkaz na druhý)
- **Dědičná:** jeden objekt přejímá vlastnosti prvního (například artefakt dle meta artefaktu)
- **Transformační:** značí především transformaci objektu za určitý čas (například dítě stárnoucí na dospělého člověka)



Obrázek 2-3: Relační konektory UUBML

Zdroj: vlastní

Oba tyto nástroje tak byly nezbytným pomocníkem při návrhu a implementaci komplexního řešení informační podpory Unicorn College, které je věnována celá následující kapitola.

### **3 Rozšíření a automatizace teritoria Unicorn College**

Vysoká škola Unicorn College používá ke své hlavní informační podpoře systém Unicorn Universe, který skýtá rozsáhlé možnosti správy a řízení běhu školy, nicméně tyto možnosti je teprve třeba implementovat. Z tohoto důvodu došlo k vytvoření zadání pro dva projekty, jež mají usnadnit a automatizovat práci v systému.

Prvním projektem je „Publikace studijních výsledků“, jež měl být nasazen pro použití jak pro UCL, tak pro společnost Top Gun Academy (dále TGA), která se zabývá školením pracovníků.

Druhým projektem je „Automatizace zápisu předmětů“, což je komplexní řešení zápisu předmětů do akademického ročníku, kdy má student možnost si jednoduše zvolit požadované předměty a jedním stiskem tlačítka si je všechny zapsat, bez nutnosti dohledu či práce studijního oddělení.

Oba projekty byly řešeny týmem Unicorn Universe for Vladimírovo (dále UU4V), který je součástí společnosti Unicorn Solutions. Zadání bylo předáno k řešení na podzim 2012 v září. Od této doby jsem v rámci půl roku pod vedením týmu UU4V navrhl, vytvořil skripty, implementoval, otestoval a předal klientovi finální řešení, včetně potřebných dokumentací a prezentací nového řešení.

#### **3.1 Publikace studijních výsledků**

##### **3.1.1 Analýza současného stavu**

V Unicorn College až do letního semestru 2011/2012 neměl student svoji Kartu studenta v předmětu a veškerá hodnocení byla zaznamenávána do artefaktu *Studijní výsledky*, který měl každý student pouze jeden. V rámci něho listy reprezentovaly studijní semestr. Na každém listu byly studijní výsledky ze všech předmětů, které student v daném semestru studoval.

Proces publikace byl řešen prostřednictvím třech klientských skriptů. První skript, využívající UES API, shromáždil hodnocení předmětů, které následně shellový skript zkonvertoval z ODS do CSV. Poslední skript posléze hodnocení zaktualizoval. Tento postup publikace výsledků byl tedy příliš centralizovaný na několik kompetentních osob, rigidní a generoval velké množství chyb. Rovněž byl artefakt se studijními výsledky často nepřehledný, obzvláště pokud shrnoval velké množství studovaných předmětů.

V TGA nebyla publikace výsledků nijak automatizována, veškerá hodnocení byla ukládána do ODS souborů a ručně aktualizována v editoru na příslušné Karty účastníků. Účastníkům kurzu tak mohly být studijní výsledky prezentovány i s týdenním zpožděním a samotná aktualizace výsledků zabrala příliš zbytečné repetitivní práce, při které vznikalo spousty chyb vlivem lidského faktoru.

### **3.1.2 Implementace projektu**

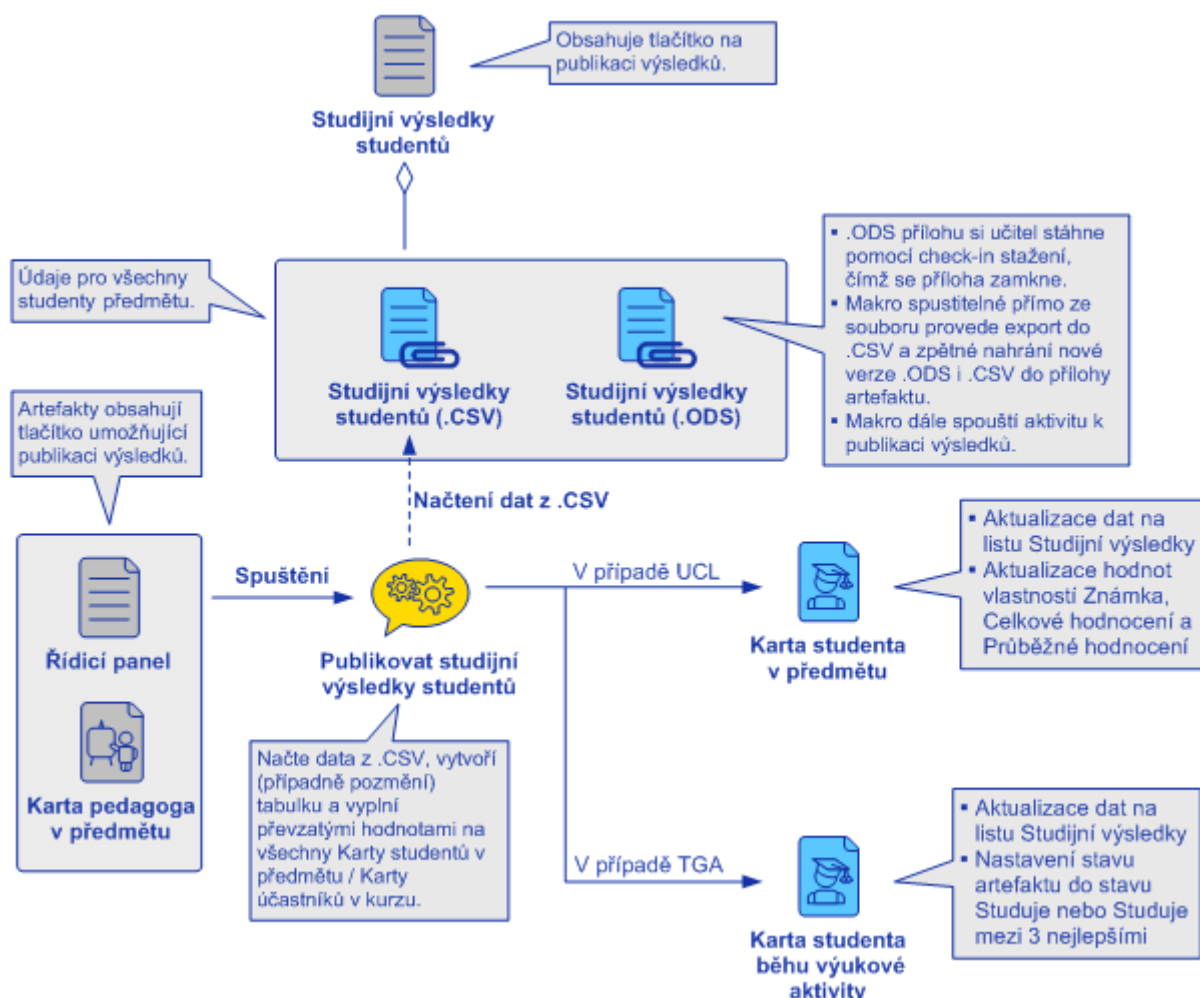
Každému studentovi, který se do předmětu zapíše řádným způsobem, je zakládán artefakt *Karta studenta v předmětu* (v TGA se tento artefakt nazývá *Karta studenta běhu výukové aktivity*). Student má díky tomuto artefaktu ucelený přehled o předmětu, o důležitých sděleních, domácích úkolech k řešení, absolvovaných testech a o studijních výsledcích.

Na tento artefakt jsou rovněž publikovány výsledky studijních aktivit (testy, domácí úkoly, docházka apod.), průběžné a celkové hodnocení a známka. Pedagog má možnost evidovat studijní výsledky v souboru v příloze artefaktu Studijní výsledky studentů.

Mezi hlavní myšlenky tohoto řešení patří decentralizace možnosti publikovat výsledky na jednotlivé předměty či kurzy a celkové usnadnění procesu. V každém předmětu i kurzu je tedy na jeden semestr zakládán artefakt Studijní výsledky, čili správu hodnocení má v kompetenci každý lektor.

Ve společnosti TGA je, jak již bylo zmíněno, každému účastníkovi kurzu automaticky zakládán artefakt Karta studenta běhu výukové aktivity. Tento druh artefaktu se od Karty studenta v předmětu liší především životním cyklem a cílenou skupinou uživatelů, proto

musel být proces publikace studijních výsledků optimalizován a generalizován pro použití v obou situacích.



Obrázek 3-1: Schéma procesu publikace výsledků

Zdroj: vlastní

Na obrázku Obrázek 3-1: Schéma procesu publikace výsledků je vyobrazeno komplexní schéma procesu publikace, jež se skládá ze dvou důležitých částí.

Tou první je klientské makro spouštěné v programu Libre Office Calc, což je tabulkový procesor operující s buňkami podobný konkurenčním programům Microsoft Office Excel nebo Open Office Calc, přičemž struktura vstupního souboru je k vidění na obrázku Obrázek 3-2: Struktura vstupního souboru s hodnocením. Soubor je ve formátu ODS.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Nazev predmetu kódy predmetu		Průběžné hodnocení					Závěrečné hodnocení			Zkouška (1. termín)			Zkouška (2. termín)		
2	Příjmení a jméno	UID	Test 1	Domácí úkol 1	Bonus	Docházka	Průběžné hodnocení celkem	Zkouška	Hodnocení celkem	Známka	Body	Datum	Zkoušející	Body	Datum	Zkoušející
3	Medián		21	4	#VALUE!	7	31	0	47	4						
4	Arnold Jaku	11-1	36	4		7	47	0	47	4	0	17.2.2012	Tesař Marek	0	17.2.2012	Tesař Marek
5	Bělik Adam	1-1-1	5	3		6	14									
6																
7																
8																
9																
10																
11																

Obrázek 3-2: Struktura vstupního souboru s hodnocením

Zdroj: vlastní


Hlavním účelem tohoto makra je export požadovaných listů (identifikovaných dle názvu listu) do datově nenáročného formátu CSV, neboli Comma-separated Values. Všechny tyto soubory jsou následně nahrány jako příloha na artefakt Studijní výsledky studentů, kde dojde k založení skriptové aktivity, která slouží ke spouštění skriptu běžící na straně serveru.

Toto makro kombinuje skriptovací jazyk JavaScript a technologii Representational State Transfer pro Application Programming Interface (dále API) informačního systému Unicorn Universe. Tato technologie umožňuje vzdáleně volat služby UU pomocí HTTP protokolu, čímž je například vytvoření přílohy na artefaktu (konkrétní příklad kódu je vidět v příloze A).

Další částí projektu byl skript běžící na straně serveru sloužící k vygenerování tabulek se studijními výsledky (příklad studijních výsledků je vidět na obrázku Obrázek 3-2: Struktura vstupního souboru s hodnocením). Tento skript nejprve načte na artefaktu Studijní výsledky studentů CSV přílohy, z nichž získá data o studentech v předmětu a jejich hodnocení, která použije k vytvoření tabulky se studijními aktivitami dle vzoru na obrázku Obrázek 3-3: List studijní výsledky.

Skript, běžící na straně serveru, je napsaný v objektově orientovaném jazyce Ruby, jehož ukázka je k vidění v příloze B. Tento skript nejprve ze vstupních CSV příloh načte studijní aktivity, studenty, jejich hodnocení a také mediány výsledků daných studijních aktivit. Dále dopočítá pořadí studentů v každé studijní aktivitě, vytvoří tabulku s výsledky a buňky s výsledky podbarví dle legendy (viz obrázek 3-3). Dále také doplní hodnoty vlastností Průběžné hodnocení, Celkové hodnocení, Znamka a Aktualizováno dne. V případě

spuštění skriptu pro TGA dojde k nastavení stavu Karty studenta běhu výukové aktivity do stavu Studuje nebo Studuje mezi třemi nejlepšími.

<a href="#">Základní informace</a>   <a href="#">Studijní výsledky</a>   <a href="#">Domácí úkoly</a>   <a href="#">Absolvované testy</a>			
<b>Studijní výsledky</b> 			
Známka	2		
Průběžné hodnocení	270		
Celkové hodnocení	354		
Aktivita	Výsledek	Pořadí	Medián
Test 1	30	28	45
Test 2	66	19	63
Domácí úkol 1	2	42	44
Test celkem	98	37	
Domácí úkol 2	7	40	48
Bonus	26	32	46
Docházka	41	25	51
Průběžné hodnocení celkem	270	42	434
Zkouška	84	12	50
Hodnocení celkem	354	39	491
Aktualizováno dne: 19.03.2013 16:10:23			

**Požadavky k absolvování**

Známka	Počet bodů
1	175 - 200
2	150 - 174
3	120 - 149
4	0 - 119

**Minimální požadavky**

Typ hodnocení	Počet bodů
Průběžné hodnocení	60
Zkouška (maximum 100 bodů)	60

**Legenda**

Výsledek účastníka je **NAD** mediánem výsledků dané aktivity

Výsledek účastníka je **POD** mediánem výsledků dané aktivity

Výsledek účastníka je **SHODNÝ** s mediánem výsledků dané aktivity

Obrázek 3-3: List studijní výsledky

Zdroj: vlastní

## 3.2 Automatizace zápisu předmětů do akademického roku

### 3.2.1 Analýza současného stavu

V minulých letech byly zápisy studentů řešeny operativně servisními zásahy, vyžadující napsání dočasných klientských skriptů a spousty ruční práce. Stávající způsob byl založen zejména na hromadném zpracování dat z požadavků k zápisu ve formě CSV souborů,



jejich následnou modifikací a hromadnou aktualizací dat v Unicorn Universe. Tento způsob řešení byl administrativně i časově náročný, data byla často v systému promítnuta s velkým zpožděním.

Důležitým předpokladem pro automatizace je konzistence informací v Unicorn College, které je naštěstí dosaženo, a to jak v oblasti evidence studentů, tak v oblasti jednotlivých běhů předmětů.

V rámci řešení projektu bylo manipulováno s artefakty typu Karta studenta, Elektronický index či Role studenta, a naštěstí všechny tyto artefakty měli konzistentní kódovou konvenci, čili bylo možné je velice snadno dohledávat podle UID. Zároveň se jejich agregované objekty lišily v minimální míře.

Současně tak i běhy předmětů byly s pevně danou a dodržovanou organizační strukturou. V kódu organizační jednotky předmětu byly vždy tři znaky vyjadřující předmět (např. MA1 znamená Matematika 1), a také ročník a semestr. Tímto kódem se pak řídili všechny podřízené artefakty, jako je studijní skupina nebo portál předmětu. Automatizace tak byla implementovatelná i bez složitějších zásahů do struktury UCL.

### **3.2.2 Implementace projektu**

Každý akademický rok je zahajován řádným zápisem předmětů, které si student přeje absolvovat. V kreditovém systému Unicorn College si student na začátku zimního semestru zapisuje předměty na celý akademický rok, přičemž k výběru má jakýkoliv předmět, nezávisle na studovaném oboru a ročníku. Student pouze musí absolvovat předměty povinné ke svému studijnímu oboru.

Nedílnou součástí jsou i korekce zápisu, které probíhají dvakrát ročně a díky nimž si student může provést změnu zapsaných předmětů, například si dodatečně předmět zapsat nebo jeho zápis zrušit, nebo změnit studijní skupinu. Tyto korekce probíhají na začátku zimního i letního semestru.

V rámci projektu bylo nutno řešit několik zásadních problémů:

- Vysoká zátěž systému – Většina studijních informačních systémů běžně používaná ve školství (STAG, KOS apod.) trpí nestabilitou a dlouhou dobou odezvy v okamžiku spuštění zápisu předmětů. Vysoký nápor uživatelů přihlašujících se a zapisujících si předměty v krátkém časovém intervalu je velice podobný DDoS útoku (Distributed Denial of Service), kdy je server přehlčován dotazy vedoucími k jeho zastavení, ne-li vypnutí.
- Nedostatečný přehled o kapacitách – Studenti na Unicorn College neměli možnost snadno zjistit, jaké studijní skupiny jsou v daném předmětu volné a zda jejich kapacita není již naplněná. Z tohoto důvodu bylo nutné integrovat systém, který by studentům poskytl aktuální přehled. Zároveň bylo potřeba zajistit snadnou správu tohoto přehledu.
- Paralelní souběh skriptů – V projektu se také muselo předpokládat vysoký počet studentů snažících se zapsat v jeden okamžik, čímž by docházelo k souběhu skriptů na objektech společných pro celý předmět, jako je seznam studentů nebo aktuální naplnění studijní skupiny.
- Nedostatečné oprávnění studentů – Každý student má v systému pouze jednu pracovní studentskou roli, která mu neposkytuje dostatečná oprávnění k manipulaci s většinou potřebných objektů, například student ve své studentské roli nemá právo editovat svůj elektronický index nebo vytvářet Kartu studenta v předmětu. Toto omezení bylo nutné obejít, aby mohl být zápis bez dohledu nadřazených rolí.
- Termíny zápisu – Doposud nebyl v Unicorn Universe integrován automatizovaný systém zápisu předmětů, začátek i konec zápisu byl tedy určen pouze rozhodnutím studijních administrátorů, kdy přestanou zapisovat. Tento proces bylo tedy nutné automatizovat, aby reagoval na stanovené termíny, ale přesto dovolil nadřazeným rolím spustit zápis kdykoliv dodatečně pro opoždilé studenty.
- Obecné řešení – Z důvodu veliké diverzity jednotlivých předmětů bylo nutné skripty generalizovat, aby byly schopné reagovat na všechny standardní i alternativní scénáře. Například předmět Tělocvik má jiné studijní formy (tím je myšleno prezenční nebo kombinovanou formu studia) než předmět Projekt v Androidu, stejně tak může být předmět vyučován česky, anglicky, nebo oběma jazyky.

### 3.2.3 Zápis předmětů

Tato část projektu se týká formálního aktu zápisu předmětů na celý akademický rok. Jedná se o důležitý prvek zahájení studia, ve kterém si student může jednoduše vybrat požadované předměty k absolvování a stiskem tlačítka je zapsat.

Součástí této části řešení je řídicí panel jednotný pro všechny semestry a všechny ročníky, na kterém jsou umístěna tlačítka umožňující komplexní správu celého procesu zápisů, k vidění je na obrázku Obrázek 3-4: Řídicí panel zápisu předmětů.

The screenshot displays a web interface for subject registration, organized into several functional panels:

- Termíny (Dates):** A section for setting registration dates. It includes input fields for 'Aktuální ročník' (set to '<Aktuální ročník>') and 'Aktuální semestr' (set to 'Zimní'). Below are date pickers for 'Začátek zimního zápisu', 'Konec zimního zápisu', 'Začátek zimní korekce', 'Konec zimní korekce', 'Začátek letní korekce', and 'Konec letní korekce'. An 'Uprav hodnoty' button is at the bottom.
- Generování artefaktů k zápisu (Generating artifacts for registration):** Contains instructions to update the student list and a 'Hromadné generování' button. A note mentions generating artifacts individually for 1 to 5 students, with a corresponding button.
- Export Karet studentů (Export student tickets):** Shows 'Vybrané typy stavů Karty studenta:' with three status options: 'Požadovaný stav 1: Počáteční', 'Požadovaný stav 2: Aktivní', and 'Požadovaný stav 3: Aktivní alternativní'. A 'Generuj seznam studentů' button is present.
- Správa běhu předmětů (Subject run management):** Features two buttons: 'Aktualizace portálů' (described as updating the student list on portals) and 'Aktualizace studijních skupin' (described as updating group capacity artifacts).
- Zápis do letního semestru (Registration for the summer semester):** A section with explanatory text and a 'Spust letní zápis' button.

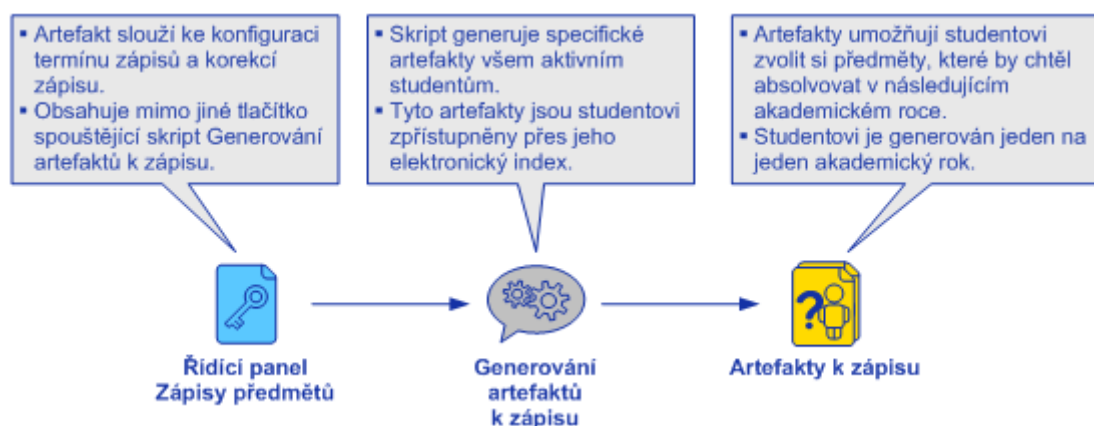
Obrázek 3-4: Řídicí panel zápisu předmětů

Zdroj: vlastní

Zápis předmětů probíhá ve třech fázích:

1. Fáze *Generování artefaktů k zápisu* je spouštěná centrálně koordinátory výuky na řídicím panelu. Každému studentovi je předběžně vygenerován artefakt k zápisu sloužící jako zápisový list, na kterém je seznam všech předmětů rozdělených dle ročníků.
2. Fáze *Zápis do zimního semestru* je individuální a závislá pouze na studentovi. V této fázi je studentovi zaktivněn artefakt k zápisu, student si vybere požadovaný předměty, studijní skupiny a tlačítkem si je zapíše.
3. Fáze *Zápis do letního semestru* je naopak centrálně spouštěná z řídicího panelu. Student již musí mít zvolené požadované předměty do letního semestru (případně si je může zapsat dodatečně v letní korekci), ale v momentě zapisování v zimním semestru ještě předměty do letního semestru nemusí mít určené studijní struktury, kapacity, vytvořené portály předmětu apod.

První fáze *Generování artefaktů k zápisu* zajišťuje značnou úlevu na zatížení systému, neboť je možné tuto operativní přípravu rovnoměrně rozložit na delší časový úsek. Nicméně je to podmíněno aktuálností seznamu studentů, což se dá snadno zařídit tlačítkem v tabulce Export karet studentů na řídicím panelu dle obrázku Obrázek 3-4: Řídicí panel zápisu předmětů. Toto tlačítko vytvoří CSV přílohu, jejímž obsahem jsou klíčová data o studentech, jejich jméno, Universe ID, forma, jazyk studia a ročník. Tato CSV příloha následně slouží jako zdroj pro serverový skript generující specifické artefakty ke studiu, závislé dle formy a jazyka studia.



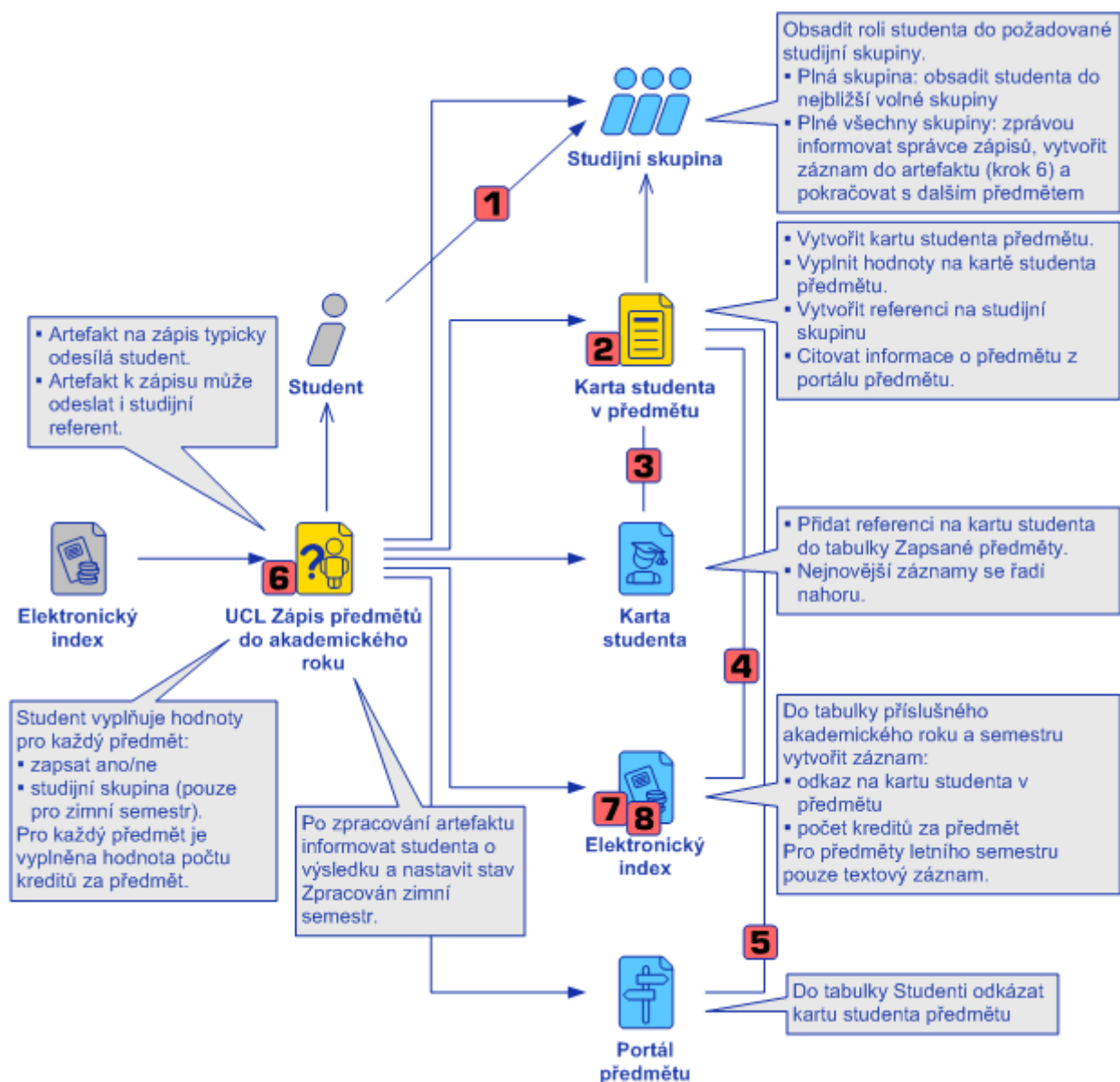
Obrázek 3-5: Schéma generování artefaktů k zápisu

Zdroj: vlastní

V druhé fázi *Zápis do zimního semestru* probíhá finální registrace předmětů dle schématu na obrázku Obrázek 3-6: Schéma zápisu do zimního semestru, které je celé založené na jednom serverovém skriptu spouštěném automaticky po odeslání artefaktu k zápisu. Student z postavení své studentské role v systému nemá dostatečná oprávnění na téměř žádnou z těchto operací, proto bylo nutné použít pro serverový skript prostředníka, tzv. *systémového uživatele*. Jde o speciální druh uživatele, který má v systému stejné možnosti jako běžný uživatel, nicméně je možné se na něj přihlásit pouze ze skriptů běžících na straně serveru. Tím pádem je možné tomuto systémovému uživateli poskytnout spoustu oprávnění bez ohrožení bezpečnosti a soukromí.

Tato fáze byla problémová také z důvodu možnosti křížení transakcí na jednom zdroji, například přímo při prvním kroku (viz obrázek Obrázek 3-6: Schéma zápisu do zimního semestru) je po obsazení studenta do studijní skupiny přičítána jednička do aktuálního naplnění. V případě, že by se 2 studenti najednou zapsali do stejné studijní skupiny v jeden okamžik, by buď došlo k chybě a skript by se ukončil, nebo by došlo k doplnění špatné hodnoty. Z tohoto důvodu byla na zdroje, u nichž je krátká doba manipulace (v řádu jednotek vteřin), použita komponenta Zámek. Tato komponenta dokáže zamknout libovolný zdroj (soubor, artefakt, agregovaný objekt) na lokální úrovni, znamenající, že uživatel přes grafické rozhraní má k danému zdroji stále přístup, nicméně skripty ne. V případě paralelního souběhu skriptů má přednost ten, který daný zdroj zamknul dříve, druhý proces čeká na odemčení tím způsobem, že se pokouší zdroj zamknout vlastním zámkem, přičemž čekací doba mezi jednotlivými pokusy uzamčení se exponenciálně zvyšuje až do určité maximální doby.

Nicméně tato varianta je aplikovatelná pouze na objekty, které jsou zamykány lokálně. Bohužel z logiky zápisu může docházet k souběhu skriptů na portálu předmětu při editování seznamu studentů, který je umístěn na agregovaném objektu typu list. Listy jsou zamykány na perzistentní úrovni limitující i přístup uživatele přes grafické rozhraní a neumožňující stabilní čekání na odemčení. Z tohoto důvodu byl tento krok centralizován do jednoho skriptu spouštěného pravidelně v noci, který seznamy studentů postupně aktualizoval a předešel tak křížení transakcí skriptů.



Obrázek 3-6: Schéma zápisu do zimního semestru

Zdroj: vlastní

Poslední fází je *Zápis do letního semestru*. Jde již o termínově neřízený zásah, který je rovněž možné spustit z řídicího panelu tlačítkem. Časová nezávislost umožňuje optimalizace, z tohoto důvodu nejprve řídicí skript této fáze naplánuje spuštění skriptů k zápisu předmětů do letního semestru na každém artefaktu k zápisu s určitým časovým rozestupem. Tím se omezí možnost souběhu skriptů a dojde k rovnoměrnému rozložení zátěže na delší časový úsek.

### 3.2.4 Korekce zápisu

Další neméně důležitou částí projektu automatizace zápisu předmětů v Unicorn College jsou korekce zápisu. Každý student má dodatečně možnost jak ovlivnit své zapsané předměty i po tom, co vyplní a jednorázově odešle artefakt k zápisu. Subjektem korekce zápisu je vždy jeden zvolený předmět, který je možné si dodatečně zapsat, změnit studijní skupinu nebo si zápis zrušit.

Technické řešení nezahrnuje jakoukoliv nutnost zásahu studijního oddělení či studijních referentů, student je závislý tedy pouze na svých rozhodnutích a omezený pouze vypsanými termíny ke korekci zápisu pro zimní a letní semestr (korekce zápisu na Unicorn College tedy probíhají dvakrát ročně).

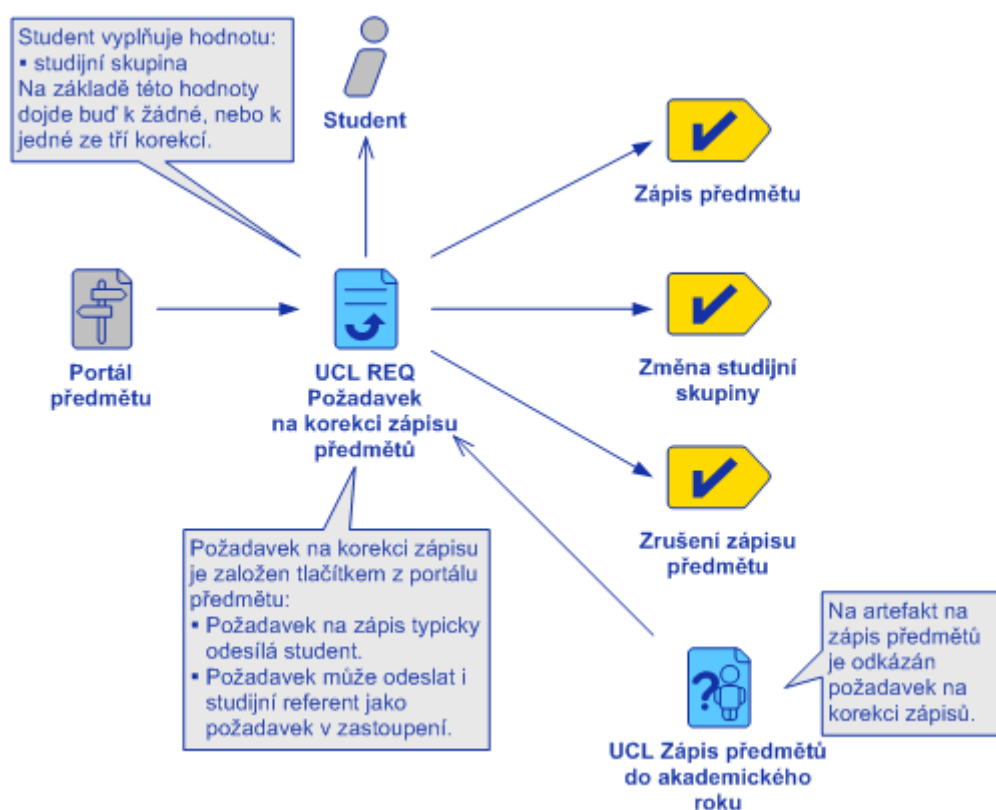
Samotný proces je spouštěn nad hlavním rozcestníkem informací o předmětu, na portálu, na kterém se nachází tlačítko zakládající artefakt speciální třídy požadavek. Na tomto požadavku ke korekci má student možnost zvolit si požadovaný typ korekce, tedy si student vybírá ze seznamu možností zahrnující výběr zápisu do studijní skupiny či zrušení zápisu předmětu. Po výběru typu korekce je možné požadavek odeslat ke zpracování, které je plně automatizované skripty běžícími na straně serveru. Student se pouze dozví výsledek korekce zápisu informační aktivitou ve svém úkolovníku.

Projekt bylo nutné také připravit na možné alternativní scénáře použití, například může dojít k pokusu zapsat se do neexistující studijní skupiny. Pokud by k tomuto došlo, serverové skripty korekci ukončí a informují studenta o jeho chybě. Bohužel jde o omezení ze strany informačního systému Unicorn Universe, neboť neumožňuje dynamicky generovat formuláře. Seznam nabízených forem korekce je tedy statický výčet obsahující možnosti Zapsat do studijní skupiny 1 – 4 a Odepsat z předmětu. Z tohoto důvodu bylo nutné ošetřit následující alternativní scénáře:

- Zápis do neexistující studijní skupiny
- Zápis do studijní skupiny, ve které již je student zapsán
- Zrušení zápisu předmětu, na kterém student není zapsán
- Zápis do studijní skupiny, jejíž maximální kapacita byla naplněná

Dalším alternativním scénářem je také snaha o korekci zápisu mimo vymezené termíny, která je rovněž hlídána serverovými skripty navrženými tak, aby studijní oddělení mělo možnost kdykoliv spustit korekci zápisu v zastoupení za studenta. K tomuto je využita vlastnost třídy artefaktu Požadavek, neboť autor požadavku je jedním ze základních identifikátorů požadavku. Pokud je autorem požadavku role, jejíž meta artefakt odpovídá studentské roli, tak je závaznost termínu korekce kontrolována.

Na obrázku Obrázek 3-7: Schéma korekcí zápisu je shrnut proces společný všem třem druhům korekce. Každý individuální požadavek je možné použít pouze k jedné korekci zápisu, v případě potřeby je však možné snadno spustit korekci znovu na novém požadavku. Tímto jsou separovány informace o provedených korekcích na jednotlivé požadavky a je například velice snadno zjistitelný důvod neúspěchu korekce.

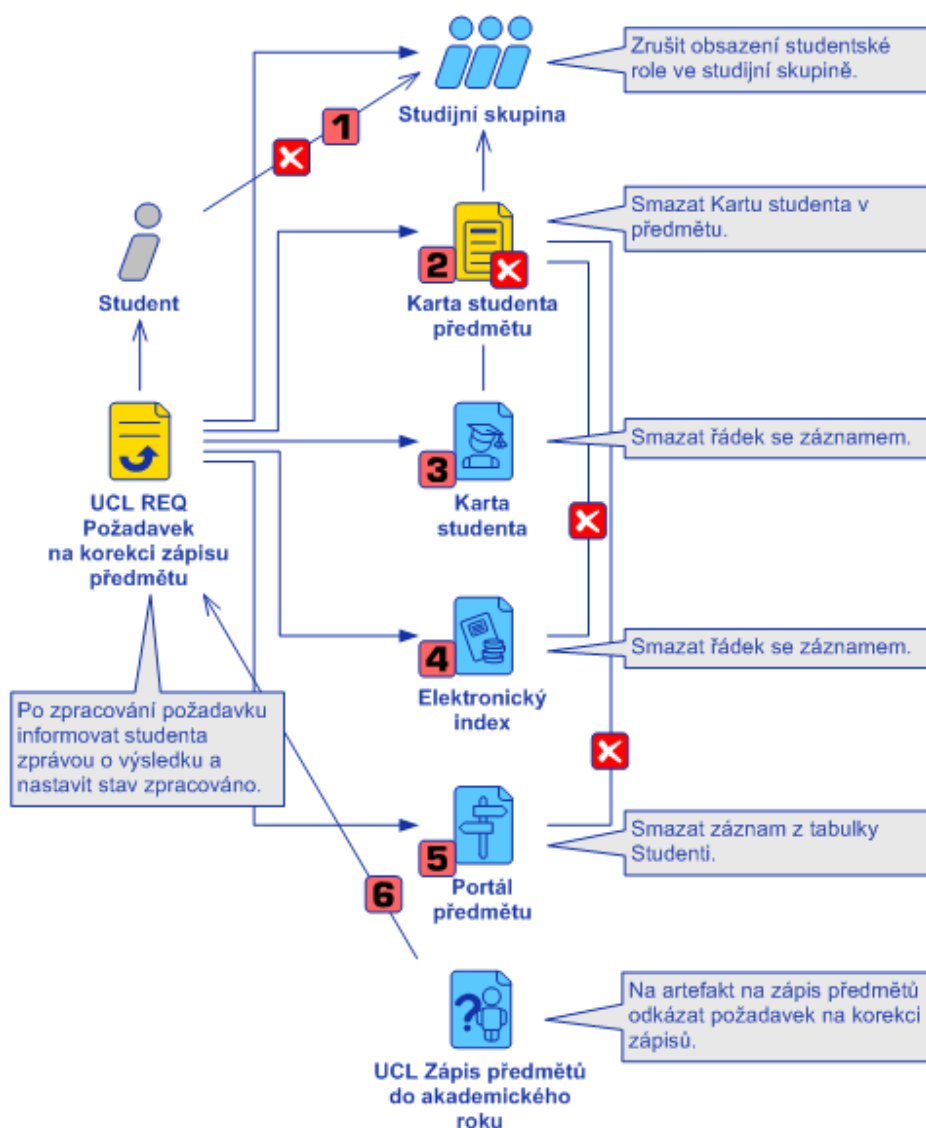


Obrázek 3-7: Schéma korekcí zápisu

Zdroj: vlastní



Níže na obrázku Obrázek 3-8: Schéma zrušení zápisu předmětu jsou k vidění jednotlivé kroky v případě zrušení zápisu předmětu. V případě regulérní korekce zápisu dochází ke zrušení obsazení ze skupinové role studijní skupiny, ke smazání artefaktu Karta studenta v předmětu, k odstranění referencí na studenta ze seznamu zapsaných studentů na portálu předmětu a k odstranění referencí na předmět ze seznamu studovaných předmětů na elektronickém indexu a kartě studenta.



Obrázek 3-8: Schéma zrušení zápisu předmětu

Zdroj: vlastní

## 4 Zhodnocení přínosu

### 4.1 Publikace studijních výsledků

Projekt publikace studijních výsledků je prvním z vyřešených projektů a je v současné době schválen klientem a nasazen do ostrého provozu, díky tomu jsou získávány první odezvy spolehlivosti a účinnosti.

Samotné vyhotovení projektu stálo zhruba 400 hodin úsilí čistého pracovního času, které zahrnovalo úpravy metodických artefaktů, tvorbu skriptů, testování a komplexní přípravu business teritoria Unicorn College k používání tohoto řešení. Náklady k implementaci tohoto projektu jsou k vidění v tabulce 1, nicméně jde pouze o hrubý odhad, neboť informace o platech jsou brány z průměrných hodnot z portálu [www.platy.cz](http://www.platy.cz).

Tabulka 1 – Nákladnost projektu publikace studijních výsledků

Pozice	Průměrný plat	Počet hodin	Celkem [Kč]
IT konzultant	241,49	175	42 261
IT Projektový manažer	302,48	75	22 686
Programátor	202,23	150	30 335
			95 282

Zdroj: Platy.cz, 2013

Níže v tabulce 2 jsou pro porovnání odhadnuté náklady pro manuální publikaci studijních výsledků za každý měsíc. Tyto odhady se ovšem opět nemusí blížit skutečnosti, neboť celkový čas se může velice lišit. Pro porovnání, manuální publikace studijních výsledků pro 10 studentů by trvala cca 20 minut. Po implementaci projektu je možné tento čas zmenšit až na desetinu, v případě rostoucího počtu studentů až na setinu. Návratnost se tedy pohybuje mezi 3 – 6 měsíci.

Tabulka 2 – Náklady na manuální publikaci výsledků

Pozice	Průměrný plat	Počet hodin	Celkem [Kč]
Lektor	137,38	30	<b>4 121</b>
Vysokoškolský pedagog	148,73	30	<b>4 462</b>
Asistent učitele	74,71	60	<b>4 483</b>
Specialista technické podpory IS	171,82	15	<b>2 577</b>
			<b>15 643</b>

Zdroj: Platy.cz, 2013

Tento projekt tedy vede nejen k nesmírné úspoře času, ale i k obecnému zlepšení informovanosti studentů a zvýšení vizuální přehlednosti. Každý student je nyní ve svém snažení porovnáván s ostatními, neboť je ke každé studijní aktivitě dopočítáno pořadí a znázorněna střední hodnota.

## 4.2 Automatizace zápisu předmětů

Tento projekt byl náročnější svým rozsahem zhruba o polovinu více než publikace studijních výsledků a byl cílen pouze pro užití Unicorn College. Projekt není přenositelný, nejde o instalovatelnou aplikaci, neboť je přímo omezen a sestaven pro užití v informačním systému Unicorn Universe, jenž škola využívá ke své administraci i výuce.

Nasazení projektu probíhalo ve dvou fázích. Vzhledem k tomu, že zápis předmětů do letního semestru a zároveň s ním i letní korekce zápisu byla naplánována na únor 2013, bylo nutné se zaměřit především na stihnutí tohoto termínu, přičemž automatizace zápisu předmětů bude komplexně využita až v září a říjnu 2013, kdy bude probíhat zápis do celého akademického roku 2013/2014.

Řešení bylo tedy navrženo tak, aby bylo snadno upravitelné k dosažení maximální účinnosti i po skončení letního semestru. K zápisu předmětů do letního semestru byla využita data ze září 2012, kdy si studenti rovněž volili předměty, jež by chtěli absolvovat. Letní zápis byl tedy omezen o použití artefaktů k zápisu, neboť bylo organizačně velice náročné donutit studenty znovu si zvolit požadované předměty. Korekce zápisu však byla

nasazena téměř kompletní, pouze nedochází ze zřejmých důvodů k referenci provedené korekce na studentův artefakt k zápisu.

Náklady na tento projekt jsou shrnuty v tabulce 3, ačkoliv jde opět pouze o hrubý odhad, mohou se tak lišit od skutečnosti v řádu desítek tisíců.

Tabulka 3 – *Nákladnost projektu automatizace zápisu předmětů*

<b>Pozice</b>	<b>Průměrný plat</b>	<b>Počet hodin</b>	<b>Celkem [Kč]</b>
IT konzultant	241,49	250	<b>60 374</b>
IT Projektový manažer	302,48	125	<b>37 810</b>
Programátor	202,23	225	<b>45 502</b>
			<b>143 685</b>

Zdroj: Platy.cz, 2013

Pokud by se ovšem UCL rozhodlo provést zápis předmětů do akademického roku a s tím související korekci zápisu pomocí servisní jednotky, tak by se jim tento jednorázový zásah příliš nevyplatil, viz tabulka 4. V této tabulce je porovnána odhadovaná náročnost každého zimního i letního zápisu předmětů, pokud by byla řešena operativně za běhu zápisů. Tento způsob řešení by sice vyšel levněji, nicméně jde o jednorázový zásah a hned při dalším zápisu předmětů by se investice do projektu vrátila.

Tabulka 4 – *Náklady na servisní zápis předmětů*

<b>Pozice</b>	<b>Průměrný plat</b>	<b>Počet hodin</b>	<b>Celkem [Kč]</b>
Systémový administrátor	199,94	150	<b>29 991</b>
Správce informačního systému	190,85	100	<b>19 085</b>
Specialista technické podpory IS	171,82	200	<b>34 364</b>
Programátor	202,23	100	<b>20 223</b>
			<b>103 663</b>

Zdroj: Platy.cz, 2013

Projekt „Automatizace zápisu předmětů“ nesmírně ulehčí práci všem pracovníkům Unicorn College a zároveň zahrnuje několik nových funkcí, jako je ucelený přehled kapacit studijních skupin, evidence provedených korekcí zápisu či automatická tvorba seznamu zapsaných studentů na předmětu. Toto vede k zlepšení image školy na trhu a zvýšení spokojenosti studentů s informační podporou výuky.

Projekt je rovněž navržen k udržení kompatibility s nadcházejícími verzemi systému Unicorn Universe i ke snadné přenositelnosti do budoucích akademických ročníků. Jediný nutný uživatelský zásah, který umožní zapsat předměty do nového akademického ročníku, je správné nastavení dvou hodnot (Aktuální semestr a Aktuální ročník) na řídicím panelu zápisu předmětů, viz obrázek 3-4.

## **Závěr**

Cílem této bakalářské práce bylo zlepšení informační podpory Unicorn College v rámci systému Unicorn Universe, konkrétně usnadnění publikace studijních výsledků a automatizace zápisu předmětů do akademického roku. S čistým svědomím mohu říci, že těchto cílů bylo dosaženo, a to v plné výši.

Oba projekty jsou nyní plně nasazeny k užití a uspokojují potřeby studentů i studijních referentů Unicorn College. Odezva klienta je rovněž pozitivní a s kladným hodnocením, i přes nastalé změny v metodice Unicorn ES Powered Company. Začátkem roku 2013 jsem byl seznámen se změnami týkajícími se této metodiky, její restrukturalizace a nastolení nových standardů. Proto je tedy možné, že se teoretická část bude lišit od aktuálních principů a postupů společnosti Unicorn a.s., nicméně projekty byly vytvořeny s dlouhodobou životností a kompatibilitou pro budoucí verze systému i metodiky.

Tyto projekty ušetří nesmírné množství času a repetitivní práce a vedou ke zvýšení spokojenosti se studiem na Unicorn college. Díky automatizačním činnostem je možné alokovat lidské síly na potřebnější úkoly, čímž Unicorn College dosahuje lepších ekonomických výsledků.

V rámci budoucích etap automatizace Unicorn College bych navrhoval například zakládání nových běhů předmětů, automatizace v rámci správy zaměstnanců školy a možnost poskytovat externí kurzy a školení.

Jak již bylo zmíněno, oba projekty jsou svojí specifíčností a vázáním na informační systém Unicorn Universe nepřenositelné. Pokud by se ovšem jiná vysoká škola rozhodla používat stejný IS, tyto projekty by bylo možné nabízet jako samostatné produkty za předpokladu podobného organizačního schématu dané VŠ, případně úpravy konfigurací projektu, což by mohlo vést k dalšímu ekonomickému zisku.

## Seznam citací

- [1] T. BERNERS-LEE, R. FIELDING a L. MASINTER. *Uniform Resource Identifier (URI): Generic Syntax* [online]. 2005 [cit. 2013-02-10]. Dostupné z: <http://tools.ietf.org/html/rfc3986>
- [2] *Unicorn ES Powered Company* [online]. 2012 [cit. 2013-02-10]. Dostupné z: <http://unicornuniverse.eu>
- [3] KOVÁŘ, Vladimír. *Unicorn Enterprise System Powered Company : Metodika pro řízení podniku a organizací s přímou podporou informačního systému*. [s.l.], 2011. 130 s. Dizertační práce. Univerzita Hradec Králové, Fakulta informatiky a managementu, Katedra informatiky a kvalitativních metod.
- [4] *Platy.cz: přehled platů, průměrná mzda, průměrný plat* [online]. 2013 [cit. 2013-05-09]. Dostupné z: <http://www.platy.cz/>

## Seznam použité literatury

- [5] Elektronická databáze článků ProQuest [online]. 2012 [cit. 2012-10-13]. Dostupné z: <http://knihovna.tul.cz/>
- [6] GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. *Podniková informatika*. 2., přeprac. a aktualiz. vyd. Praha: Grada, 2009, 496 s. Expert (Grada). ISBN 978-80-247-2615-1.
- [7] KOCIÁN, Václav. *Automatizace vysoké školy Unicorn College nad platformou Unicorn ES*. Praha, 2011. Dostupné z: [http://www.unicorncollege.cz/katalog-bakalarskych-praci/vaclav-kocian/attachments/Posudek\\_vedouciho\\_Kocian\\_Vaclav.pdf](http://www.unicorncollege.cz/katalog-bakalarskych-praci/vaclav-kocian/attachments/Posudek_vedouciho_Kocian_Vaclav.pdf). Bakalářská práce. Unicorn College. Vedoucí práce Ing. Marek Beránek, Ph.D.
- [8] KOVÁŘ, Vladimír et al. *Unicorn ES Powered Company - Management*. 1. revidované vydání. Praha: Unicorn College, 2009. ISBN 978-80-87349-01-4.

## **Seznam příloh**

A – UKÁZKA KÓDU UŽIVATELSKÉHO MAKRA

B – UKÁZKA RUBY SERVEROVÉHO SKRIPTU



## Příloha A – Ukázka kódu uživatelského makra

```
1102 // get attachmentlist
1103 if (i == 0) { // stačí zavolat jednou, art je vždy stejný
1104
1105     var errorNumber = "401";
1106     while (errorNumber == "401") {
1107         response = httpRequest(subjectUri, "GET",
1108             "attachment/UESAttachment/getAttachmentList", null,
1109             password);
1110         // zjistíme výsledek z odpovědi, zajímá nás hlavně číslo
1111         // chyby
1112         // např. HTTP/1.1 404 Not Found
1113         errorNumber = response.substring(9, 12);
1114         if (errorNumber == "401") {
1115             showInfoMessageBox(pUp, "Chyba autentizace",
1116                 "Uživatel nebyl autentizován.Zadejte znova přihlašovací údaje.");
1117             var password = dialogBox();
1118         } else if ((errorNumber == "500")) {
1119             throw "Uživatel se nepřihlásil správně.";
1120         } else if ((errorNumber != "200") && (errorNumber != "401")) {
1121             throw "REST volání UES API zhlásilo výjimku, více viz odpověď serveru:\n\n"
1122                 + response;
1123         }
1124     }
1125
1126     // je tam ODS? jsou tam CSV? Jsou zamčené?
1127     var jsonString = response.substring(response.indexOf("{"),
1128         response.lastIndexOf("}") + 1);
1129     var jsonResponse = JSON.parse(jsonString);
1130
1131     var totalSize = jsonResponse.totalSize;
1132     var jsonPage = jsonResponse.pageEntries;
1133
1134 }
```

## Příloha B – ukázka Ruby serverového skriptu

```
364 subjects.each do |subj_code, subj_info|
365   # nejprve načteme kód předmětu
366   m = subj_code.match(/^{\w{3}}\d{2}(S|W)$/)
367   subj_code_short = m[1]
368   subj_info[:name] = get_subject_name(subj_code_short)
369
370   # pokud jde o předmět letního semestru, zápis probíhá pouze v podobě zápisu do logu
371   if m[2] == "S"
372     subj_info[:sg_info] = SG_INFO[form_sym][:ok_s]
373     @registered_subjects << "Předmět #{subj_code_short} #{subj_info[:name]} zapsán do letního semestru."
374     next
375     # pokud není vyplněná vlastnost určující požadovanou skupinu pro zimní semestr, tak chyba
376   elsif subj_info[:req_sg].to_s.empty?
377     # uložíme do hashe subjects konkrétní informaci o chybě
378     subj_info[:sg_info] = SG_INFO[form_sym][:no_req_sg]
379     @logger.warning("Předmět #{subj_code} nemá vyplněnou požadovanou studijní skupinu.")
380     @warnings << "Předmět #{subj_code} nemá vyplněnou požadovanou studijní skupinu."
381     next
382   end
383
384   subj_info[:sg], subj_info[:sg_info], subj_info[:portal] = study_group_cast(subj_code, subj_info[:req_sg])
385   # pokud se nepovede obsadit studenta z očekávatelného důvodu, skript pokračujem zápisem dalšího
386   # předmětu
387   next if !subj_info[:sg]
388
389   subj_info[:scc] = create_student_course_card(subj_code, subj_info, year)
390   # pokud již existuje karta studenta v předmětu, pokračujeme dalším předmětem
391   # pouze teoretická situace, prakticky to nastane velice vzácně
392   if !subj_info[:scc]
393     subj_info[:sg_info] = SG_INFO[form_sym][:cant_scc]
394     @logger.warning("Karta studenta v předmětu s kódem #{subj_code}/#{student[:uid]} již existuje.")
395     @warnings << "Karta studenta v předmětu s kódem #{subj_code}/#{student[:uid]} již existuje."
396     next
397   end
end
```